Homepageveröffentlichung unbefristet genehmigt für SIGNON Deutschland GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt von DVV Media Group, 2018.

Verifikation und Validierung von ausführbaren Spezifikationsmodellen

The verification and validation of executable specification models

Oliver Lemke

In immer mehr Entwicklungsprojekten für bahntechnische Systeme kommen Spezifikationen auf Basis ausführbarer Anforderungsmodelle zum Einsatz. Das damit einhergehende Versprechen höherer Spezifikationsqualität und der einfachen Nachnutzung von Modellen lässt sich jedoch nur einlösen, wenn diese hohe Qualitätsanforderungen erfüllen. In diesem Beitrag wird ein systematisches Vorgehen zur Verifikation und Validierung von Anforderungsmodellen vorgestellt, um diesen Qualitätsanspruch sicherzustellen.

1 Aktueller Stand

Seit längerer Zeit werden in Entwicklungsprojekten der Leit- und Sicherungstechnik (LST) modellbasierte Methoden angewendet. Erste Anwendungsfälle lagen im Bereich der Systementwicklung, z. B. durch Softwaremodelle in der Beschreibungssprache Unified Modeling Language (UML [1]). Seit einigen Jahren kommen Spezifikationsmodelle ebenfalls verstärkt zum Einsatz, um die Anforderungen der Betreiber zu definieren. Dabei wird beispielsweise die Systems Modeling Language (SysML [2]) verwendet, die eine Erweiterung der UML darstellt. Kennzeichen einer modellbasierten Spezifikation ist, dass Anforderungen nicht als einfacher, natürlich-sprachlicher Text vorliegen, sondern als ein maschinenverständliches Abbild in einem Modellierungswerkzeug. Ist das Modell reich genug an Informationen, so kann man es je nach Werkzeug auch ausführen, d. h., das zukünftige Verhalten eines Systems auf Anforderungsebene simulieren zu können, bevor es entwickelt oder gebaut wurde. In der SysML ist dies z. B. durch Modelle auf Basis von Zustandsmaschinen möglich. Da ausführbare Modelle viele Vorteile bieten (Simulierbarkeit, Testbarkeit, erweiterte Nachnutzung), wird im Rahmen dieses Artikels angenommen, dass die hier angesprochenen Modelle immer auch ausführbar sind.
Mit der Nutzung von Modellen ist üblicherweise die Erwartung verknüpft, die Qualität der Anforderungsspezifikation hinsichtlich

- Eindeutigkeit
- Korrektheit
- Verständlichkeit

zu verbessern. Diese drei Aspekte sollen insbesondere helfen, Fehler und Widersprüche in der Anforderungsspezifikation möglichst früh im gesamten Entwicklungsprozess aufzudecken, da dann die Fehlerbeseitigungskosten vergleichsweise gering sind.

2 Herausforderungen

Problematisch ist an der oben formulierten Erwartung, dass die Qualitätsansprüche an die Anforderungsspezifikation mit der Prä-

More and more development projects for railway systems are making use of specifications based on executable requirement models. The ensuing promise of higher quality specifications and the easy reutilisation of such models, however, can only be kept, if these models meet high quality requirements. This article describes a systematic approach to the verification and validation of requirement models which are capable of ensuring this required quality.

1 The current state

Model-based approaches have been used in development projects for signalling and control systems (LST) for some time now. The initial use cases were in the area of system development, namely software models using the Unified Modelling Language (UML [1]). In recent years, specification models have also been increasingly used to define operators' requirements. An example of this is the System Modelling Language (SysML [2]), which is an extension of the UML. The hallmark of a model-based specification lies in the fact that the requirements do not exist as simple, natural-language texts, but as a machine-understandable image of the requirements in a modelling tool. If the model provides enough information, it can be executed according to the tool. This means that it is able to simulate the future behaviour of a system at the requirement level even before such system has been developed or built. In SysML, for example, this is can be achieved using models based on state machines. Given that executable models offer many advantages (simulation capability, testing capability and extended reutilisation), it has been assumed within the context of this article that the models mentioned herein are always executable.
The use of these models usually comes with the expectation of improved quality in the requirement specification with regard to

- unambiguity
- accuracy
- intelligibility

It is expected that these three aspects will assist in the detection of errors and inconsistencies in the requirement specification early on in the development process, when the costs for fixing such errors are still comparatively low.

2 The challenges

The problem with the aforementioned expectation lies in the fact that the quality demands for the requirement specifica-

Homepageveröffentlichung unbefristet genehmigt für SIGNON Deutschland GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt von DVV Media Group, 2018.

zision des verwendeten Beschreibungsmittels wachsen: Während eine natürlich-sprachliche Spezifikation noch Unschärfebereiche und Freiheitsgrade enthalten kann, repräsentiert ein ausführbares SysML-Modell ein exakt deterministisches Verhalten, welches auch im Fall von objektiv vorhandenen Fehlern keinen Interpretationsspielraum lässt. Um es umgangssprachlich auszudrücken: Ist das exakt spezifizierte Verhalten in einem Modell funktional falsch, ist es eben sehr präzise funktional falsch.

Weiterhin erlauben modellbasierte Methoden die effiziente Weiternutzung der einmal erstellten Spezifikationsmodelle. Beispielsweise kann das Anforderungsmodell des Betreibers auch als Basis für Testfälle der späteren Systemabnahmetests herangezogen werden. Dabei potenziert sich jedoch auch der oben formulierte Qualitätsanspruch, da sich ein Fehler im Anforderungsmodell durch die Nachnutzung in andere Entwicklungsphasen weiterverbreitet. Daher ist es erforderlich, für modellbasierte Anforderungsspezifikationen besondere Qualitätssicherungsmaßnahmen umzusetzen, die der oben geschilderten Situation und den Eigenschaften von Modellen gerecht werden. Dies betrifft insbesondere die Abbildung des Systemverhaltens, welches durch klassische Reviews nur unzureichend zu prüfen ist.

In diesem Artikel werden Lösungsvorschläge für die folgenden Fragenkomplexe vorgestellt:

- Welcher Prozess bietet sich für die Qualitätssicherung modellbasierter Anforderungsspezifikationen an?
- Welche bewährten Werkzeuge existieren für die Qualitätssicherung von modellbasierten Anforderungsspezifikationen?

3 Einordnung in den CENELEC-Prozess

Unabhängig von den verwendeten Spezifikationsmethoden gilt für sicherheitskritische LST-Systeme die CENELEC-Normenfamilie. In der DIN EN 50126 werden dabei die relevanten Lebenszyklusphasen definiert. Die Abfolge und Abhängigkeiten der Phasen entsprechen dabei dem V-Modell. Ein sehr vereinfachtes Abbild dieses V-Modells ist in Bild 1 dargestellt. Der Entwicklungsprozess beginnt links oben mit der Spezifikation des Systemkonzepts durch den Betreiber (Operator). Dieser verfeinert die Spezifikation entlang der linken Seite des V (Specification phase), bis nach Abschluss der Phase 4 die Systemanforderungen an den Hersteller (Supplier) übergeben werden. Dieser implementiert in den Phasen 5 bis 9 sein Produkt (Implementation phase). Er übergibt es nach Phase 9 an den Betreiber, der zunächst prüft, ob das System tatsächlich seinen Anforderungen entspricht (Acceptance phase). Der Betreiber nutzt schließlich das System, hält es instand und entsorgt es nach Ende der Lebenszeit (Operation phase, aus Übersichtsgründen nicht dargestellt).

Im V-Modell lassen sich zudem sehr anschaulich zwei wesentliche Ansätze zur Qualitätssicherung zeigen, auf welche sich in der CENELEC-Normenwelt an vielen Stellen bezogen wird:

- Verifikation – prüft, ob das Ergebnis einer Phase korrekt in der Folgephase umgesetzt wurde
- Validierung – prüft, ob die Implementierung gegenüber der Spezifikation korrekt ist

Beide Methoden sind ebenfalls schematisch in Bild 1 durch Pfeile dargestellt. Modellbasierte Anforderungsspezifikationen lassen sich in dem vereinfachten V-Modell in die Specification phase einordnen, weswegen diese in Bild 1 farblich hervorgehoben ist. Die Specification phase wird im Folgenden detaillierter betrachtet. Es wird dabei – stellvertretend auch für andere Beschreibungsmittel – angenommen, dass die ausführbaren Modellanteile als Zustandsmaschinen (State machines – STM), z. B. in der SysML, realisiert sind. Die Spezifikationserstellung lässt sich zunächst in die

tion increase as the accuracy of the used methodology increases. While a natural-language specification may still contain a margin of uncertainty and degrees of freedom, an executable SysML model represents an exact deterministic behaviour, which, even in the presence of objective errors, leaves no room for interpretation. In layman's terms this means that, if the exactly specified behaviour in a model is functionally wrong, it is very precisely functionally wrong.

Furthermore, model-based methods afford the efficient reutilisation of an already created specification model. For example, the operator's requirement model can also be used as the basis for test cases for later system acceptance tests. However, the quality demands formulated above also increase even further, since any error in the requirement model will propagate when used in any subsequent development phases. It is therefore necessary to implement special quality assurance measures for model-based requirement specifications which comply with the situation described above as well as the characteristics of models. This particularly involves the mapping of the system behaviour, which cannot be verified with the required level of detail by classic manual reviews.

This article examines possible solutions to the following questions:

- What process is available for the quality assurance of model-based requirement specifications?
- Which proven tools exist for the quality assurance of model-based requirement specifications?

3 Integration into the CENELEC process

Regardless of the used specification methods, safety-critical LST systems are covered by CENELEC standards. DIN EN 50126 defines the relevant lifecycle phases. The sequence and dependencies of the phases match the V-model. Fig. 1 shows a highly simplified representation of the V-model. The development process starts at the top left with the operator's specification of the system concept, which is successively detailed along the left leg of the V (the specification phase) until the system requirements are submitted to the manufacturer (supplier) after the completion of phase 4. The supplier implements the product during phases 5 to 9 (the implementation phase) and then submits it to the operator, who first checks whether the system meets the given requirements (the acceptance phase). The operator will then use and maintain the system, as well as dispose of it at the end of its lifecycle (the operation phase, which has not been shown for the sake of clarity).

The V-model also affords a very clear representation of two approaches regarding quality assurances, which are often referred to in the CENELEC standards:

- Verification – it checks whether the result of a phase has been correctly implemented in the subsequent phase
- Validation – it checks whether the implementation conforms to the specification

A schematic representation of both methods is shown in fig. 1 by means of the arrows. In the simplified V-model, model-based requirement specifications can be allocated to the specification phase, which is why it has been highlighted in fig. 1. The specification phase will be examined in more detail below. It is assumed (also in place of other modelling tools) that the executable model components have been realised as state machines (STM), i. e. in SysML.

Homepageveröffentlichung unbefristet genehmigt für SIGNON Deutschland GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt von DVV Media Group, 2018.

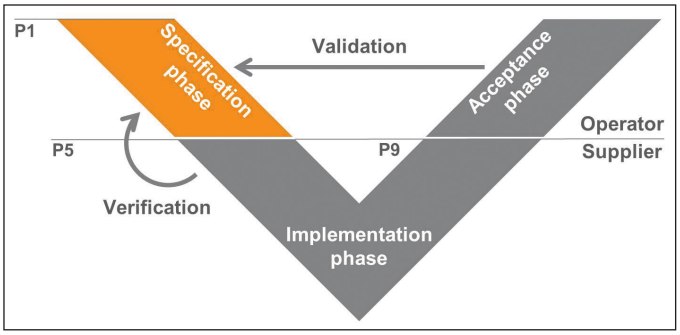

Bild 1: Vereinfachter Systementwicklungsprozess nach CENELEC
Fig. 1: A simplified system development process according to CENELEC

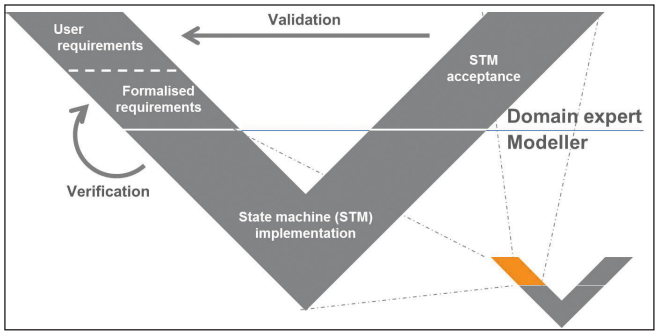

Bild 2: Inneres V-Modell zur Spezifikationserstellung
Fig. 2: An internal V-model for specification generation

folgenden vier groben Arbeitsschritte unterteilen, die in Kapitel 4.1 dann detailliert vorgestellt werden:

- Erhebung der Anwenderanforderungen – User requirements
- Formalisierung dieser Anforderungen – Formalised requirements
- Implementierung des ausführbaren Modellanteils als Zustandsmaschinen – State machine (STM) implementation
- Akzeptanz der Modellimplementierung gegen die ursprünglichen Anwenderanforderungen – STM acceptance

Das ausführbare Modell aus Zustandsmaschinen kann man dabei als eine virtuelle, abstrakte Referenzimplementierung des zukünftigen Systems auffassen. Es handelt sich bei der Erstellung eines ausführbaren Anforderungsmodells daher um einen Prozess, der Analogien zur späteren Systementwicklung aufweist. Somit liegt der Gedankengang nahe, für die Entwicklung dieses Modells genauso wie für die Entwicklung des gesamten Systems ebenfalls das V-Modell heranzuziehen. Es ergibt sich damit für die Specification phase ein weiteres, inneres V-Modell innerhalb des übergeordneten CENELEC V-Modells der Systementwicklung. Dieser Zusammenhang ist in Bild 2 dargestellt: Die oben aufgeführten Arbeitsschritte in einer typischen V-Anordnung sind erkennbar, zusätzlich ist angedeutet, wo sich das innere V-Modell zur Modellerstellung im übergeordneten V des Systementwicklungsprozesses wiederfindet. Im inneren V beginnt der Prozess oben links mit der Erhebung der Anwenderanforderungen, die dann in formalisierte Anforderungen übersetzt werden. In der folgenden Phase erfolgen dann – an der Spitze des inneren V – die Implementierungen der ausführbaren Modellanteile als Zustandsmaschinen. Anschließend muss die Akzeptanz der Modellimplementierung gegen die ursprünglichen Anwenderanforderungen geprüft werden. Ähnlich wie im übergeordneten V-Modell zwischen Betreiber und Hersteller, ergibt sich auch im inneren V eine Verteilung von Verantwortlichkeiten, hier allerdings zwischen Fachexperten für LST oder Betrieb (Domain experts) und Experten für Modellierung ausführbarer Modellanteile (Modeller). Dies ist ebenfalls in Bild 2 zu erkennen. Entscheidend ist nun, dass sich die bekannten Qualitätssicherungsansätze, Verifikation und Validierung, analog zum V-Modell der Systementwicklung auch im inneren V für die Modellerstellung anwenden lassen. Wie in Bild 2 dargestellt ist, erfolgt zunächst eine Verifikation der ausführbaren Modellanteile gegen die formalisierten Anwenderanforderungen. Nach der Erstellung des ausführbaren Modells wird dieses dann gegen die ursprünglichen Anwenderanforderungen validiert. Damit ist eine erste Antwort auf die Fragenkomplexe aus Kapitel 2 gefunden: Wird die Erstellung modellbasierter Anforderungsspezifikationen ebenfalls als Systementwicklung nach dem V-Modell betrachtet, kann die Qualitätssicherung durch die bekannten Ansätze Verifikation und Validierung erfolgen.

Initially, the generation of the specification comprises four basic steps, which have been presented in detail in section 4.1:

- The collection of the user requirements
- The formalisation of said requirements
- The implementation of the executable model components as state machines
- Acceptance of the model implementation against the original user requirements – STM acceptance.

The executable model made of state machines can be understood as a virtual, abstract reference implementation of the future system. The design of an executable requirement model is therefore a process, which bears analogies to the later system development. As such, it seems natural to use the V-model process both for the design of this model and for the development of the entire system. This provides another internal V-model for the specification phase within the higher-level CENELEC V model of system development. This correlation is shown in fig. 2. In addition to the typical V arrangement of the aforementioned process steps, this also hints at where the internal V-model for generating the model can be found in the superordinate V-model of the system development process. In the internal V-model, the process starts at the top left with the collection of the user requirements, which are subsequently translated into the formalised requirements. The next phase (at the top of the internal V) calls for the implementation of the executable model components in the form of state machines, which is followed by checking the acceptance of the model implementation against the original user requirements. Similar to the superordinate operator / supplier V-model, the internal V-model also implies a distribution of responsibilities, albeit in this case between domain experts (LST or operation experts) and modellers (modelling executable model components), as can also be seen in fig. 2. The critical question now is whether the known approaches to quality assurance, verification and validation can also be applied to the internal V-model for the purposes of modelling just like the V-model for the system development. As shown in fig. 2, the executable model components are initially verified against the formalised user requirements. Once the executable model has been developed, it will be validated against the original user requirements. This answers the questions raised in section 2: the quality assurance can be based on established verification and validation approaches, if the generation of model-based requirement specifications is interpreted as a special type of system development according to the V-model.

Homepageveröffentlichung unbefristet genehmigt für SIGNON Deutschland GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt von DVV Media Group, 2018.

4 Umsetzung des V-Prozesses zur Modellerstellung

In diesem Abschnitt wird das im vorherigen Kapitel grob beschriebene Vorgehen für die Erstellung ausführbarer, modellbasierter Spezifikationen detaillierter beschrieben.

4.1 Artefakte und Erstellungsprozess

Um die Abläufe im inneren V-Modell besser verstehen zu können, werden hier zunächst die dabei üblicherweise verwendeten Artefakte und deren Erstellung beschrieben. Eine Übersicht über die Artefakte und die Prozessschritte sowie eine Zuordnung zu den Phasen des inneren V-Modells befindet sich in Bild 3.

4.1.1 Anwenderanforderungen

Der Prozess beginnt mit der Erhebung der Anwenderanforderungen. Diese können beispielsweise als bestehende, textliche Anforderungen vorliegen oder aber als implizites Fachwissen in den Köpfen der Experten. Wenn es im Projektkontext nötig erscheint, können die Anwenderanforderungen zunächst einheitlich dokumentiert werden, z. B. in Form von Funktionslisten in einer Tabellenkalkulation.

4.1.2 Formalisierte Anforderungen

Anschließend werden die Anwenderanforderungen in eine erste formalisierte Darstellung überführt. Häufig werden dabei szenarienorientierte Beschreibungsmittel verwendet, wie z. B. die Sequenzdiagramme der SysML. Diese stellen die Kommunikation des zukünftigen Systems mit Akteuren in seiner Umwelt dar. Veranschaulicht sind die Anfragen der Akteure an das zukünftige System (Stimuli) sowie die erwarteten Antworten des Systems (Responses). Dieses Beschreibungsmittel hat sich dabei aus den folgenden Gründen als besonders geeignet erwiesen:

- Die Syntax und Semantik der Sequenzdiagramme ist gegenüber anderen Diagrammarten der SysML überschaubar und auch von Nicht-Modellierungsexperten nach kurzer Einleitung gut zu verstehen.
- Durch die Fokussierung auf die Anfragen von Akteuren und erwarteten Systemantworten und weniger auf interne Funktio-

4 The implementation of the V-process for modelling

This section describes the design process for executable, model-based specifications, which has been roughly described in the previous section, in more detail.

4.1 Artefacts and the design process

This section describes the commonly used artefacts and their generation in order to provide a better understanding of the processes in the internal V-model. Fig. 3 provides an overview of the artefacts and the process steps and their relationships with the different stages of the internal V-model.

4.1. User requirements

The process starts with the collection of user requirements, which can involve, for example, existing written requirements or the tacit knowledge of experts. The user requirements may first be systematically documented, i. e. in the form of function lists in a spreadsheet, if this is deemed necessary.

4.1.2 Formalised requirements

The user requirements are subsequently translated into an initial formalised representation. Scenario-oriented description tools, such as SysML sequence diagrams, which represent the communication of the future system with the actors in its environment, are frequently used. They illustrate the requirements of the actors with regard to the future system (stimuli) as well as the system's expected responses. This description tool has proven particularly suitable for the following reasons:

- the syntax and semantics of the sequence diagrams are clear and easily understood, even by non-modelling experts after a brief introduction, compared to other types of SysML diagrams.
- its focus on user requirements and expected system responses rather than on internal features forces developers to be requirement-driven and not to anticipate solutions.

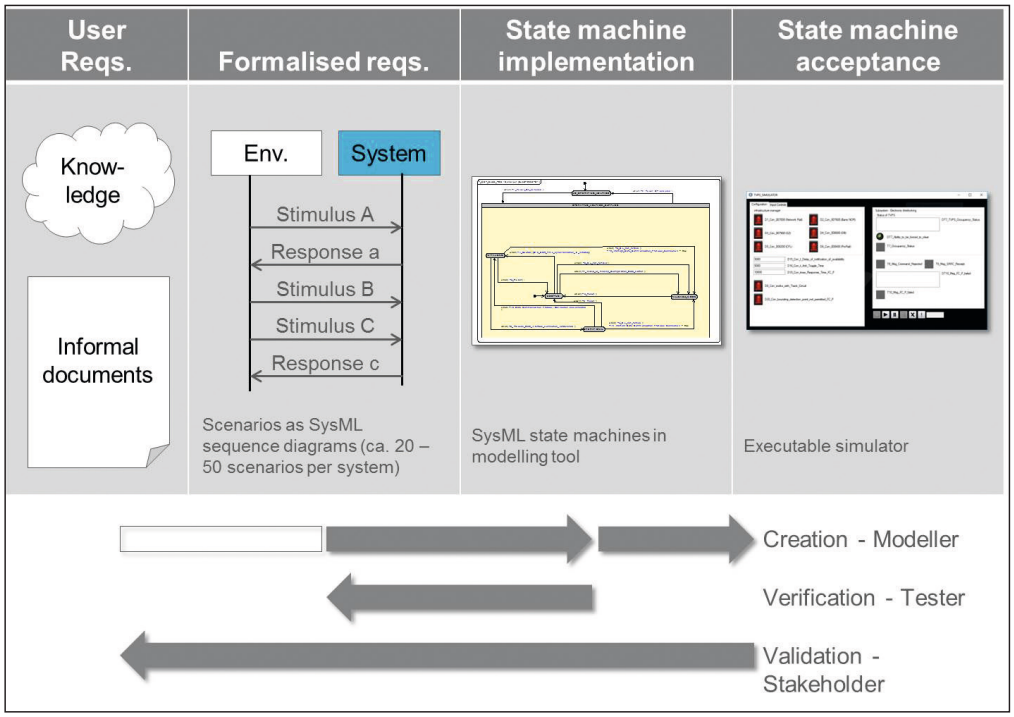

Bild 3: Artefakte und Prozessschritte zur Spezifikationserstellung
Fig. 3: The artefacts and process steps for specification generation
Quelle/*Source*: Randolf Berglehner, DB Netz AG

Homepageveröffentlichung unbefristet genehmigt für SIGNON Deutschland GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt von DVV Media Group, 2018.

nen wird der Ersteller gezwungen, anforderungsorientiert zu denken und nicht schon Lösungen vorwegzunehmen.
- Sequenzdiagramme sind ein idealer Ausgangspunkt für die spätere Erstellung von Zustandsmaschinen.

Durchgeführt wird diese Aufgabe üblicherweise gemeinsam von Fachexperten und Modellierungsexperten im Modellierungswerkzeug. Die Erfahrung hat gezeigt, dass nach einer Eingewöhnungszeit auch Fachexperten ohne Modellierungshintergrund in der Lage sind, qualitativ hochwertige Sequenzdiagramme zu erstellen.

4.1.3 Zustandsmaschinen

Um zu einem ausführbaren Modell des zukünftigen Systems zu gelangen, muss nun das zukünftige Systemverhalten durch Zustandsmaschinen modelliert werden. Diese stellen eine geschlossene Verhaltensspezifikation dar: Durch eine Zustandsmaschine ist das Verhalten eines Systems exakt und vollständig bestimmt. Wie oben schon angesprochen, handelt es sich bei diesem Schritt um eine Art abstrakte Implementierung des zukünftigen Systems. Abstrakt deshalb, weil die Implementierung auf Ebene der reinen Ablauflogik erfolgt und keinerlei technische Aspekte eines realen sicherungstechnischen Systems umfasst. Diese müssen im Modell geeignet abstrahiert werden. Das Modell muss nun derart erstellt werden, dass es sich an seinen Schnittstellen so verhält, wie es durch die formalisierten Anforderungen in Form von Sequenzdiagrammen vorgegeben wird. Anders formuliert: Wenn die Zustandsmaschine durch Stimuli entsprechend der Sequenzdiagramme angeregt wird, muss sie mit Responses ebenfalls entsprechend der Sequenzdiagramme reagieren. Dies erfordert üblicherweise tiefergehende Kenntnisse der Beschreibungsmittel und Werkzeuge, sodass diese Aufgabe in vielen Fällen durch einen dedizierten Modellierer und nicht durch die Fachexperten vorgenommen wird. Das Ergebnis ist ein ausführbares Modell innerhalb des gewählten Modellierungswerkzeuges.

4.1.4 Ausführbarer Simulator

Viele SysML-Werkzeuge erlauben es, aus dem ausführbaren Modell separat einen ablauffähigen Simulator zu erstellen. Dieser kann oftmals auch mit einer frei gestaltbaren Benutzeroberfläche versehen werden, die dann intern die Logik des ausführbaren Modells ansteuert.
Dies stellt einen erheblichen Vorteil dar, da ein solcher Simulator die Verwendung einer zur Fachdomäne passenden Benutzeroberfläche ermöglicht. Dies erhöht die Zugänglichkeit und den Nutzen des ausführbaren Modells enorm, da dieses nicht mehr nur von den Modellierungsexperten, sondern auch von den Fachexperten bedient werden kann. Mehr dazu in Kapitel 4.2.2.

4.2 Verifikation und Validierung

Nachdem das ausführbare Modell erstellt wurde, muss nun die Qualität des Modells sichergestellt werden. Wie bereits in Kapitel 3 beschrieben, lässt sich dies auch für ein Anforderungsmodell über die Ansätze zu Verifikation und Validierung umsetzen. Im Folgenden wird nun beschrieben, wie sich diese Ansätze im Kontext des inneren V-Modells konkret umsetzen lassen.

4.2.1 Verifikation

Entsprechend Kapitel 3 und Bild 2 ist das Ziel der Verifikation nachzuweisen, dass das ausführbare Modell die formalisierten Anforderungen aus den Sequenzdiagrammen erfüllt. Bei Betrachtung der beiden, an der Verifikation beteiligten Artefakte – Sequenzdiagramme als Anforderungen und ein ausführbares Modell als Implementierung – lässt sich eine weitere Analogie zur System- bzw. Softwareentwicklung ziehen:
- Die Sequenzdiagramme definieren Stimuli und erwartete Responses und entsprechen damit strukturell klassischen Testfäl-

- sequence diagrams are an ideal starting point for the later design of state machines.

This task is usually performed jointly by domain and modelling experts in the modelling tool. Experience has shown that even domain experts without a modelling background are able to design high-quality sequence diagrams after a settling-in period.

4.1.3 State machines

In order to obtain an executable model of the future system, it is necessary to model the future system's behaviour based on state machines, which provide a consistent behaviour specification. A state machine determines the behaviour of a system exactly and completely. As stated above, this step constitutes a kind of abstract implementation of the future system. It is abstract, because the implementation takes place at the level of pure processing logic and does not include any technical aspects of a real safety-related system, which needs to be suitably abstracted in the model. The model must now be generated in such a way so that it behaves at its interfaces in the manner stipulated by the formalised requirements in the form of sequence diagrams. In other words, if the state machine is stimulated by stimuli in accordance with the sequence diagrams, it must trigger the responses set out in the sequence diagrams. This usually requires deeper knowledge of the description tools. Therefore, this task is frequently performed by a dedicated modeller rather than by domain experts. The result is an executable model within the selected modelling tool.

4.1.4 An executable simulator

Many SysML tools allow the separate generation of an executable simulator based on an executable model. Often, the simulator can also be provided with a freely configurable user interface, which will internally drive the logic of the executable model.
This is a significant advantage, because such simulation models allow the use of a user interface which is suitable for the domain. This increases the accessibility and usefulness of the executable model enormously, as it allows it not only to be used by modelling experts, but also by domain experts. Section 4.2.2. contains more on this.

4.2 Verification and validation

It is necessary to verify the quality of the model once the executable model has been designed. As has already been described in section 3, the verification and validation approaches are also suitable for requirement models. The following section describes how these approaches can be implemented within the context of an internal V-model.

4.2.1 Verification

As set out in section 3 and fig. 2, the purpose of the verification process is to demonstrate that the executable model meets the formalised requirements of the sequence diagrams. Another analogy can be drawn to system or software development when considering the two artefacts involved in the verification, namely the sequence diagrams for requirements and the executable model for implementation:
- The sequence diagrams define stimuli and expected responses and are thereby structurally similar to classic test cases, which also define expected system responses to specific stimuli.

Homepageveröffentlichung unbefristet genehmigt für SIGNON Deutschland GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt von DVV Media Group, 2018.

len, die ebenfalls erwartete Systemreaktionen auf bestimmte Stimuli definieren.

- Das ausführbare Modell, insbesondere als allein ausführbarer Simulator, ist prinzipiell nichts anderes als ein Stück Software mit einem definierten Verhalten.

Aus diesen beiden Erkenntnissen kann man schließen, dass man die Verifikation von Anforderungsmodellen im beschriebenen Kontext als Software-Testproblem auffassen kann: Es geht prinzipiell darum, die Software „Simulator" mit den Testfällen „Sequenzdiagrammen" auf Korrektheit zu testen. Dazu steht am Markt nicht nur eine Vielzahl von Werkzeugen bereit, sondern auch die Prozesse sind lange bekannt und etabliert. Werden diese sinnvoll angewendet, kann die Verifikation hochgradig automatisiert und regressionsfähig ausgestaltet werden.

Ohne in diesem Artikel auf technische Details der Verifikationsdurchführung einzugehen, kann die Verifikation prinzipiell in zwei erforderliche Teilschritte zerlegt werden:

- Black-Box-Verifikation: Hierbei wird durch geeignete Testwerkzeuge überprüft, ob sich das ausführbare Modell an seinen Schnittstellen so verhält, wie es in den Sequenzdiagrammen vorgegeben ist. Falls nicht, müssen Fachexperte und Modellierer gemeinsam prüfen, woher die Verhaltensabweichung stammt. Entweder ist die Implementierung in der Zustandsmaschine oder aber die ursprüngliche Sequenz nicht korrekt.

Wichtig ist hierbei, tatsächlich nur die Konsistenz zwischen beiden Artefakten im Sinne einer Verifikation zu beurteilen. Ob das Verhalten der Zustandsmaschinen insgesamt funktional / betrieblich das gewünschte Verhalten darstellt, ist Aufgabe der Validierung.

- White-Box-Verifikation: Für die gewählten Beschreibungsmittel „Sequenzdiagramme" und „Zustandsmaschinen" ergibt sich hinsichtlich der Verifikation noch eine Besonderheit: Da es sich bei den Zustandsmaschinen um ein „reicheres" Beschreibungsmittel handelt, besteht die Gefahr, dass während ihrer Erstellung unbeabsichtigt Funktionalitäten aufgenommen werden, die in den Sequenzdiagrammen gar nicht explizit enthalten waren. Dies ist kritisch, da solch ein implizit hinzugekommenes Verhalten nicht zwangsweise den Anforderungen der Fachexperten entsprechen muss. Es muss daher explizit gemacht werden, damit eine

- The executable model, especially as a stand-alone simulation model, is basically nothing more than a piece of software with a defined behaviour.

These two findings allow us to conclude that the verification of requirement models in the described context can be regarded as a software testing problem, i. e. it is basically a matter of checking the correct operation of the "simulator" software based on the "sequence diagram" test cases. Not only is there a variety of tools available on the market, but the processes have long been known and established. The verification process can therefore be highly automated and allow for regression testing, if they are applied judiciously.

Without going into the technical details of the verification process in this article, the verification process fundamentally comprises two necessary sub-steps:

- Black Box Verification: suitable test tools check whether the executable model behaves at its interfaces as specified in the sequence diagrams. If this is not the case, domain experts and modellers must work together to determine the cause of the anomalous behaviour. Either the implementation in the state machine or the original sequence is incorrect.

The only important thing in this case is to evaluate the consistency between the two artefacts within the context of the verification. Whether the functional and operational behaviour of the state machines matches the expected behaviour is a task for the validation process.

- White Box Verification: With regard to the chosen "sequence diagram" and "state machine" description tools, the verification may entail one further problem: Since state machines are a "richer" description tool, there is a risk of the unintentional inclusion of functionalities not explicitly stated in the sequence diagrams during the design phase. This is a critical issue given that such implicitly added behaviour does not necessarily meet the requirements of the domain experts. This therefore needs to be stated explicitly, so that domain experts and modellers can coordinate a target-oriented clarification. One way to detect such behaviour is to perform a coverage analysis for the sequence

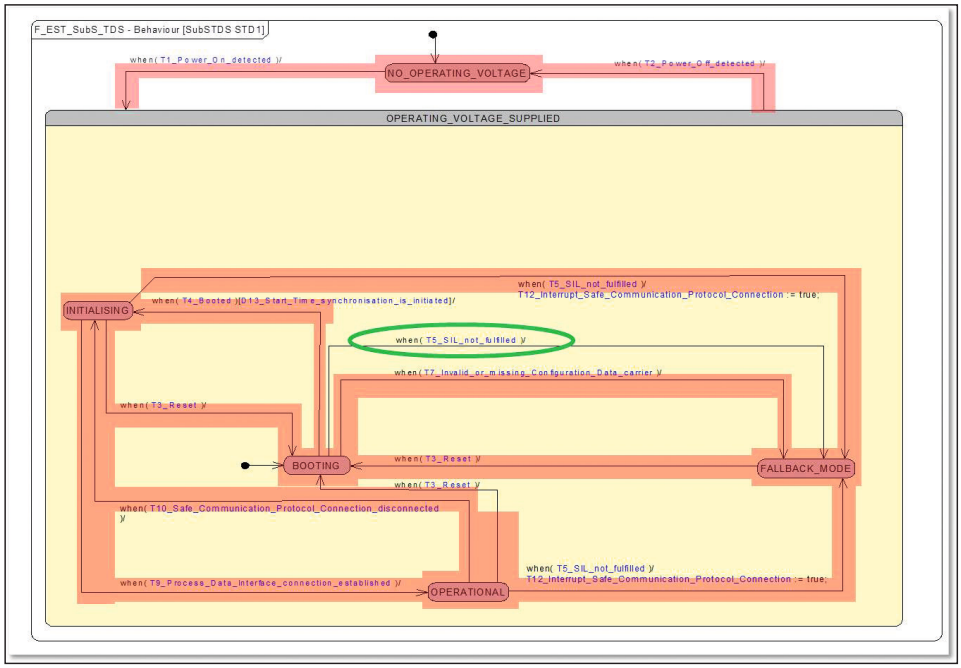

Bild 4: Überdeckungsanalyse einer Zustandsmaschine
Fig. 4: The coverage analysis of a state machine

Homepageveröffentlichung unbefristet genehmigt für SIGNON Deutschland GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt von DVV Media Group, 2018.

zielgerichtete Abstimmung zwischen Fachexperten und Modellierern erfolgen kann. Eine Möglichkeit zur Aufdeckung eines solchen Verhaltens besteht darin, eine Abdeckungsanalyse zwischen Sequenzdiagrammen und Zustandsmaschinen durchzuführen. Dabei wird jede Sequenz daraufhin geprüft, welche Zustände und Transitionen innerhalb der Zustandsmaschine angefahren werden, wenn die Sequenz abläuft. Dies kann z. B. durch eine farbliche Markierung in der Zustandsmaschine erfolgen, wie in Bild 4 dargestellt. Dort sind alle Zustände und Transitionen rot hervorgehoben, die während des Ablaufs der Sequenzdiagramme besucht wurden. Erkennbar ist eine Transition, die nicht rot markiert ist (grüne Ellipse). Hierbei handelt es sich offenbar um die Abbildung einer Funktionalität, die durch den Modellierer ohne konkrete Anforderung aus Sequenzdiagrammen hinzugefügt wurde. Die Korrektheit dieser Transition muss dann zwischen Modellierer und Fachexperte abgestimmt werden. Handelt es sich um ein gewünschtes Verhalten, muss ein zusätzliches Sequenzdiagramm definiert werden, das dieses Verhalten explizit darstellt.

Nach Durchführung dieser beiden Schritte und Abstimmung der Diskrepanzen wurde erfolgreich verifiziert, dass das Verhaltensmodell die formalisierten Anforderungen korrekt umsetzt.

4.2.2 Validierung

Ziel der Validierung ist die finale Akzeptanz des im ausführbaren Modell spezifizierten Verhaltens direkt durch die Fachexperten. Hierzu bietet es sich an, die Simulationsoberfläche des ausführbaren Modells

diagrams and state machines. Each sequence is checked to verify which states and transitions are triggered within the state machine when the sequence is run. As shown in fig. 4, this can be colour-coded in the state machine. Here, all states and transitions covered during the execution of the sequence diagram are highlighted in red. The figure shows one transition not marked in red (the green ellipse), which obviously represents a functionality added by the modeller without a specific request from the sequence diagrams. The correctness of this transition must therefore be clarified by the modeller and domain expert. Should it prove to be a desired behaviour, it is necessary to define an additional sequence diagram, which explicitly represents this behaviour.

The behaviour model was successfully verified as correctly implementing the formalised requirements after the conclusion of these two steps and the clarification of any discrepancies.

4.2.2 Validation

The purpose of validation is the final acceptance of any behaviour directly specified in the executable model by the domain experts. It therefore makes sense to use the simulation interface of the executable model to this end. As already mentioned above, it is also useful to design the user interface in such a way that it accommodates the experiences of the domain experts involved. This will facilitate both the operation and understanding of the outputs generated by the simulation model. Fig. 5 shows the user interface of a simulation model for the electrical connection of a point machine.

SAVE THE DATE!

18. Internationaler
SIGNAL+DRAHT-Kongress

08. – 09. November 2018
Fulda, Maritim Hotel

Mehr Informationen ab August unter:
www.eurailpress.de/veranstaltungen

SIGNAL+DRAHT
Kongress 2018

Homepageveröffentlichung unbefristet genehmigt für SIGNON Deutschland GmbH / Rechte für einzelne Downloads und Ausdrucke für Besucher der Seiten genehmigt von DVV Media Group, 2018.

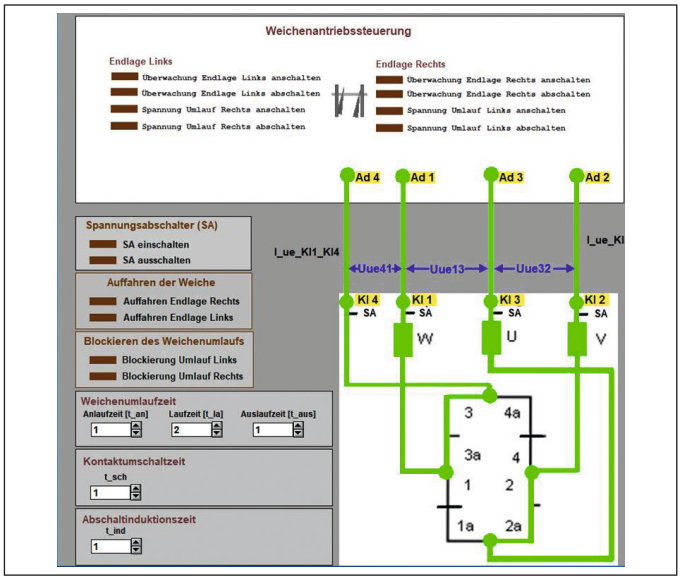

Bild 5: Simulatoroberfläche für ausführbares Weichenmodell
Fig. 5: The simulation model interface for an executable point model

heranzuziehen. Sinnvoll ist es weiterhin – wie oben schon angesprochen –, deren Benutzeroberfläche so auszugestalten, dass sie den Erfahrungen der beteiligten Fachexperten möglichst entgegenkommt. Dies erleichtert die Bedienung und das Verständnis der durch den Simulator erzeugten Ausgaben. Bild 5 zeigt die Benutzeroberfläche eines Simulators für die elektrische Anschaltung eines Weichenantriebs. Um eine möglichst hohe Unabhängigkeit in der funktionalen Abdeckung zu erzielen, sollte die Validierung nicht explizit auf Basis der Sequenzdiagramme erfolgen. Es hat sich hierbei bewährt, dass die Fachexperten in einem vereinfachten Format selbst Testfälle erstellen, die auf der ursprünglichen Struktur der Anwenderanforderungen beruhen. Wurde beispielsweise eine tabellarische Auflistung von Funktionen verwendet, wäre es sinnvoll, für jede der Funktionen mindestens einen Testfall vorzusehen. Entsprechend Bild 2 und dem oben beschriebenen Vorgehen erfolgt die Validierung unmittelbar zwischen dem ausführbaren Modell bzw. dessen Simulatoroberfläche und den Anwenderanforderungen unter Umgehung der formalisierten Anforderungen. Dies hat den zusätzlichen Vorteil, dass Fehler, die bei der Formalisierung der Anwenderanforderungen in Sequenzdiagramme eingebracht wurden, durch die Validierung aufgedeckt werden.
Die Fehler, die während der Validierung im Modellverhalten gefunden werden, müssen abgestimmt werden. Die Lösung wird dann in Sequenzdiagrammen und Zustandsmaschinen umgesetzt. Anschließend sind eine erneute Verifikation und eine erneute Validierung erforderlich.

5 Fazit

Durch die Interpretation des Erstellungsprozesses für Anforderungsmodelle als eigenen Entwicklungsprozess nach dem V-Modell lassen sich die bewährten Ansätze Verifikation und Validierung zur Qualitätssicherung heranziehen. Da es sich insbesondere bei der Verifikation lediglich um ein spezielles Software-Testproblem handelt, lässt sich dieses mit etablierten Techniken und Prozessen lösen. Somit wird eine Erstellung qualitativ hochwertiger Anforderungsmodelle möglich. In Zukunft wird jedoch verstärkt zu prüfen sein, ob nicht auch Methoden zur formellen Verifikation von Modellen auf Ebene eines mathematischen Beweises praxistauglich werden. ■

In order to achieve the highest possible independence in the functional coverage, the validation should not explicitly rely on the sequence diagrams. It has proved positive for domain experts to design their test cases themselves using a simplified format, while basing them on the original structure of the user requirements. If, for example, the functions are listed in a spreadsheet, it would be useful to plan at least one test case for each function. The executable model or its simulation interface and the user requirements are validated directly in accordance with fig. 2 and the procedure described above, while bypassing the formalised requirements. This has the added advantage that any errors introduced during the formalisation of the user requirements into the sequence diagrams are detected during validation.
Any errors detected during the model behaviour validation need to be clarified. The solution is then implemented by way of the sequence diagrams and state machines. A renewed verification and validation are subsequently required.

5 Conclusion

Interpreting the requirement model generation process as an independent development process based on a V-model allows the use of the proven verification and validation approaches for quality assurance. Given that verification merely involves a special software testing problem, it can be resolved using established techniques and processes. This enables the generation of high-quality requirement models. In the future, however, it will become increasingly necessary to examine whether the methods for the formal verification of the models have practical applicability at the level of a mathematical proof. ■

LITERATUR | LITERATURE

[1] Object Management Group (Hrsg.): OMG® Unified Modeling Language® (OMG UML®), Version 2.5.1, Dezember 2015
[2] Object Management Group (Hrsg.): OMG Systems Modeling Language, Version 1.5, Mai 2017

AUTOR | AUTHOR

Dr. Oliver Lemke
Senior Expert
SIGNON Deutschland GmbH
Anschrift /*Address*: Schützenstraße 15-17, D-10117 Berlin
E-Mail: oliver.lemke@signon-group.com