

# Zustandsbasierte Kollaborationsdiagramme

Dipl.-Ing. (FH) Cornelia Heinisch, MSc – STZ Softwaretechnik Esslingen

Zustandsbasierte Kollaborationsdiagramme, reaktive und transformatorische Systemanteile, Modellierung der Dynamik, Konsistenz von Zustandsdiagrammen und Interaktionsdiagrammen

Zur Beschreibung des dynamischen Verhaltens eines Systems werden nach UML Zustands- und Interaktionsdiagramme verwendet. Während ein Zustandsdiagramm das Verhalten eines Objektes über mehrere Anwendungsfälle hinweg beschreibt, wird im Interaktionsdiagramm die Abarbeitung eines Anwendungsfalls in der Zusammenarbeit von Objekten dargestellt. Als Interaktionsdiagramme stehen in UML Kollaborations- und Sequenzdiagramme zur Verfügung, die verschiedene Repräsentationen derselben Interaktionen darstellen. Interaktions- und Zustandsdiagramme zeigen unterschiedliche, sich ergänzende Sichten auf ein und dasselbe System. Die Modellierung eines Systems in unterschiedlichen Sichten birgt jedoch immer die Gefahr in sich, dass diese Sichten zueinander inkonsistent werden. Besonders fehleranfällig ist hierbei die getrennte Betrachtung von Objektzuständen in den Zustandsdiagrammen und den Objektinteraktionen in den Interaktionsdiagrammen. Die Konsistenz der Modellierung der Dynamik kann jedoch sichergestellt werden, indem Interaktionen und Zustände abgestimmt aufeinander in den „Zustandsbasierten Kollaborationsdiagrammen“ in einem Guss betrachtet werden. Diese elegante Kombination von Zustandsdiagramm und Kollaborationsdiagramm erlaubt in einer einzigen Sicht die vollständige Beschreibung des reaktiven und transformatorischen Verhaltens der Objekte bei der Abarbeitung eines Anwendungsfalls und stellt damit die Konsistenz von Objektzuständen und Objektinteraktionen sicher.

## 1 Einleitung

Zur Modellierung von Systemen in Systemanalyse und Systementwurf stehen verschiedenste Diagrammarten der UML zur Verfügung [1], [2], [3], [4]. Jedes Diagramm betrachtet dabei einen speziellen Aspekt des Systems. Kollaborations<sup>1</sup>- und Zustandsdiagramme modellieren nach UML dynamische Abläufe des Systems. Obwohl sowohl in Kollaborations- als auch in Zustandsdiagrammen die Dynamik des Systems beschrieben wird, gibt es einen Unterschied: Während sich Zustandsdiagramme dazu eignen, das Verhalten eines Objekts über mehrere Anwendungsfälle hinweg zu beschreiben – hierfür sind

---

<sup>1</sup> Kollaborationsdiagramme, die eine Nummerierung der Reihenfolge der Nachrichten enthalten, und Sequenzdiagramme beinhalten die gleiche Information und können ineinander umgerechnet werden.

Kollaborationsdiagramme ungeeignet – eignen sich Kollaborationsdiagramme dazu, einen speziellen Anwendungsfall durch die Zusammenarbeit von Objekten zu modellieren. Zustandsdiagramme und Kollaborationsdiagramme stellen also zwei sich ergänzende Sichten auf ein und dasselbe System dar. Bei der Verwendung von zwei getrennten Sichten auf ein und dasselbe System muss jedoch stets sichergestellt werden, dass die Beschreibungen in den unterschiedlichen Sichten insgesamt wieder konsistent sind. Zum Beispiel muss gewährleistet werden, dass bei der Abänderung einer Kollaboration zwischen Objekten auch die zugehörigen Zustandsautomaten der betroffenen Objekte angepasst werden.

Werden Kollaborationsdiagramme und Zustandsdiagramme in den „Zustandsbasierten Kollaborationsdiagrammen“ abgestimmt aufeinander in einem gemeinsamen Diagramm betrachtet, so ist es viel einfacher, die Systembeschreibung konsistent zu halten. Dies liegt daran, dass bei der Änderung einer Kollaboration zwischen Objekten die Auswirkungen auf die Zustandsautomaten der betroffenen Objekte sofort sichtbar werden. Entsprechend werden Auswirkungen einer Änderung eines Zustandsautomaten in der Kollaboration der Objekte sofort sichtbar. In den neuen „Zustandsbasierten Kollaborationsdiagrammen“ kann damit das reaktive<sup>2</sup> Verhalten der Objekte bei der Abarbeitung eines Anwendungsfalls vollständig und konsistent beschrieben werden, wobei zugleich das transformatorische<sup>3</sup> Verhalten spezifiziert wird.

In Kapitel 2 wird die herkömmliche Modellierung der Dynamik mit Kollaborations- und Zustandsdiagrammen dargestellt und im Vergleich dazu werden in Kapitel 3 die neuen „Zustandsbasierten Kollaborationsdiagramme“ vorgestellt.

## 2 Modellierung der Dynamik in getrennten Sichten

Bild 2-1 zeigt ein Kollaborationsdiagramm der Systemanalyse. Es wird hierbei die Kollaboration von Objekten in einem Anwendungsfall „Pannenmeldung durchführen“ dargestellt. Der Anwendungsfall ist Teil eines Systems „Ferndiagnose Kraftfahrzeug“. In diesem Beitrag ist jedoch das genaue Verständnis des Ablaufs dieser Anwendung nicht erforderlich.

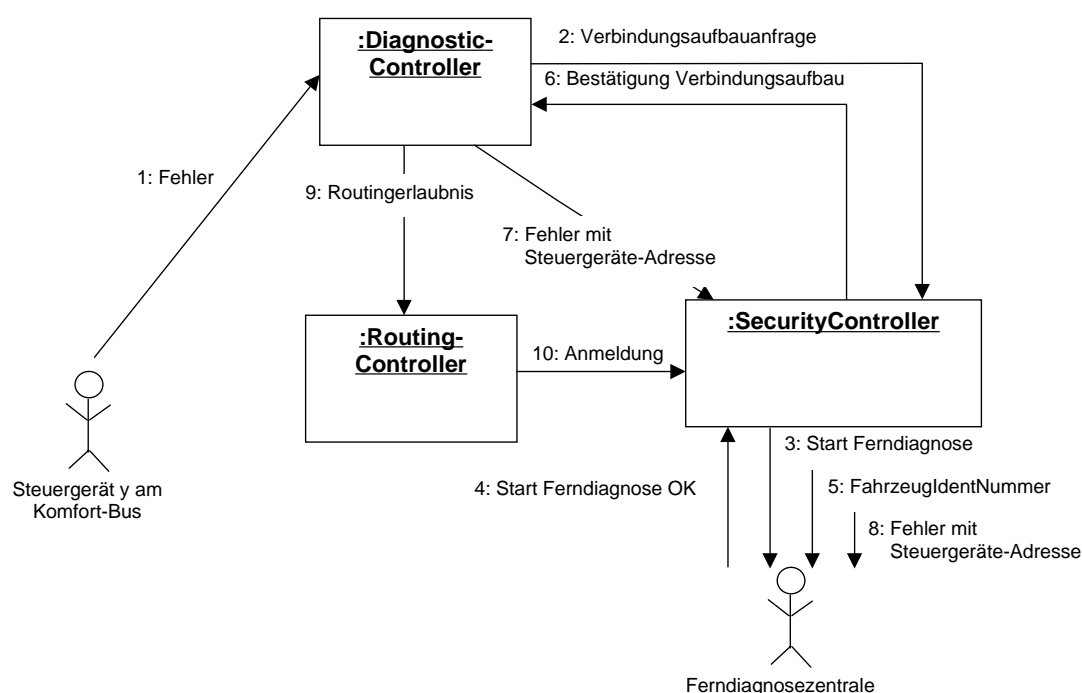


Bild 2-1 Kollaborationsdiagramm der Systemanalyse

<sup>2</sup> Ein reaktives System reagiert auf die Ereignisse seiner Umgebung, wobei die Reaktion vom jeweiligen Systemzustand abhängen kann.

<sup>3</sup> Ein transformatorisches System berechnet einen Output aus einem Input, wobei der Output nur vom Input abhängt und nicht von einem Zustand des Systems.

Sollen zusätzlich zu den Objektinteraktionen auch die Objektzustände betrachtet werden, kann nach UML ein Zustandsdiagramm für die einzelnen Objekte gezeichnet werden. Bild 2-2 zeigt das Zustandsdiagramm für das Objekt vom Typ `DiagnosticController`. Da es sich hier um ein Zustandsdiagramm der Systemanalyse handelt, werden nur die Zustände und Transitionen der Verarbeitungslogik dargestellt. In einem Zustandsdiagramm des Systementwurfs müssten alle Zustände und Transitionen des technischen Systems<sup>4</sup> vollständig beschrieben werden.

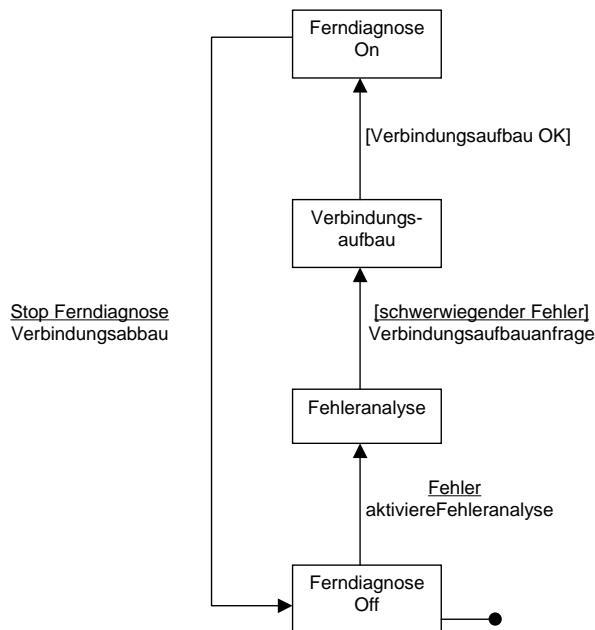


Bild 2-2 Zustandsdiagramm der Systemanalyse

Der Zusammenhang zwischen den Objektinteraktionen im Kollaborationsdiagramm und den Zuständen und Transitionen im Zustandsdiagramm lässt sich nur mit Mühe herstellen, obwohl beide Diagramme für sich betrachtet richtig und vollständig sind. So kann aufgrund der beiden Diagramme angenommen werden, dass sich das Objekt vom Typ `DiagnosticController` im Zustand „Ferndiagnose Off“ befindet, bevor der Fehler (siehe Nummer 1: in Bild 2-1) gemeldet wird. Weiterhin lässt sich ableiten, dass die ankommende Nachricht mit der Nummer 6, „Bestätigung Verbindungsaufbau“ (siehe Bild 2-1), mit der Bedingung [Verbindungsaufbau OK] zu verknüpfen ist. Generell gibt es aber keine Regel, wie die Zustände und Transitionen mit den Objektinteraktionen des Kollaborationsdiagramms zusammenhängen. Erschwerend kommt hinzu, dass bei jeder Änderung in der Kollaboration – der Fehler wird beispielsweise nicht mehr durch den `DiagnosticController` analysiert, sondern durch den `RoutingController` – auch die Zustandsdiagramme umzuzeichnen sind. Da aber die Objektinteraktionen nicht mit den Transitionen verknüpft sind, ist es gar nicht so einfach zu sehen, welcher Teil des Zustandsautomaten in ein anderes Objekt zu verlagern ist. Diese Probleme können umgangen werden, wenn die Zustände und Transitionen gemeinsam mit den Objektinteraktionen einer Kollaboration in einem „Zustandsbasierten Kollaborationsdiagramm“ betrachtet werden.

### 3 Modellierung mit „Zustandsbasierten Kollaborationsdiagrammen“

In einem „Zustandsbasierten Kollaborationsdiagramm“ sind die objektinternen Zustände und die Transitionen in Abhängigkeit von den Objektinteraktionen sichtbar. Bild 3-1 zeigt ein Zustandsbasiertes Kollaborationsdiagramm für den in Kapitel 2 betrachteten Anwendungsfall „Pannemeldung durchführen“. Dabei sind nur die Zustandsdiagramme der Objekte vom Typ `DiagnosticController` und `RoutingController` eingeblendet. Ein Ereignis IN x, das einen Zustandsübergang bewirkt, bedeutet, dass die Nachricht mit der Nummer x einen Zustandsübergang bewirkt. Eine Aktion OUT x, die bei einem Zustandswechsel ausgeführt wird, bedeutet, dass die Nachricht mit der Nummer x versandt wird.

<sup>4</sup> In einem technischen System befasst man sich über die Verarbeitungslogik hinaus auch mit technischen Aspekten wie zum Beispiel Parallelität und Fehlerhandling.

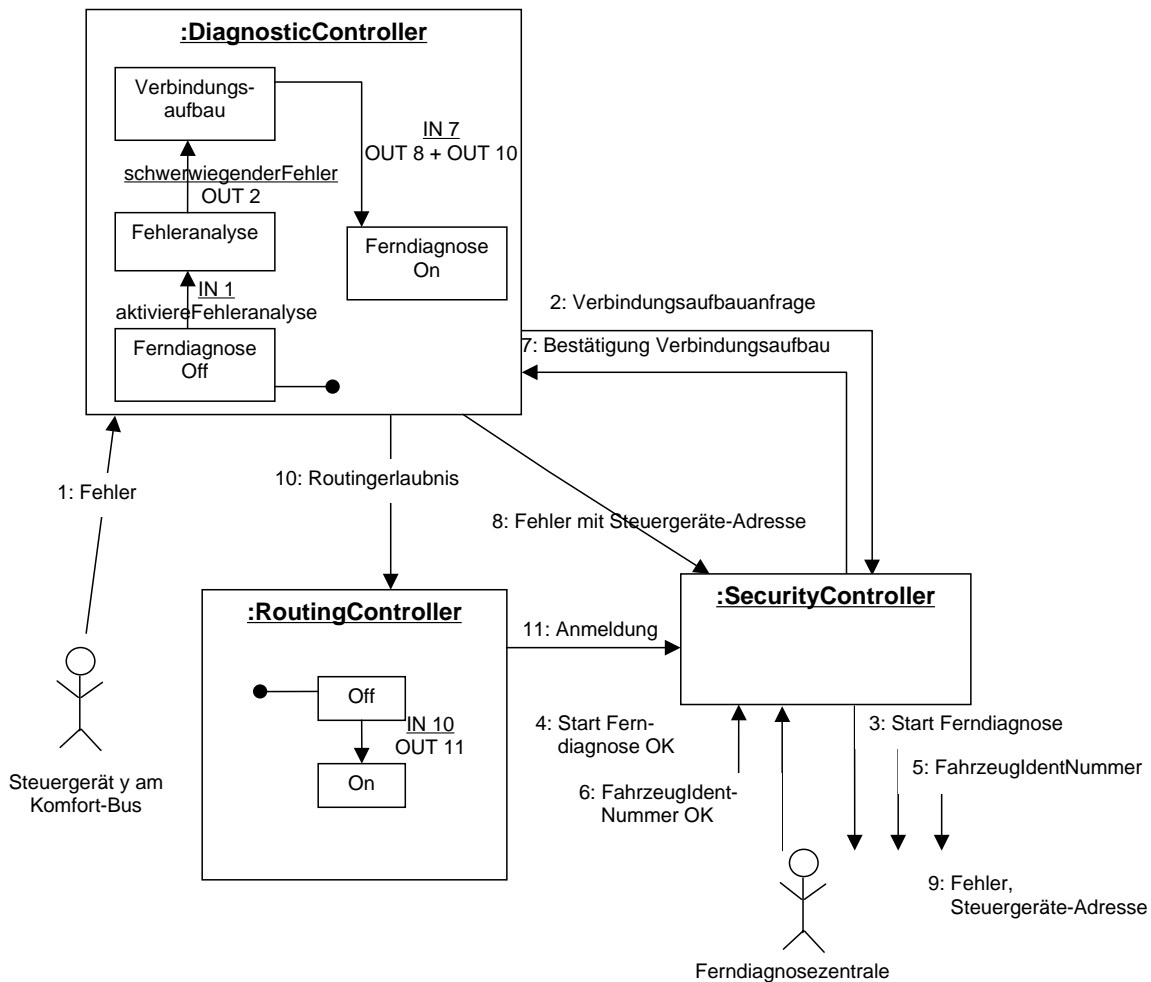


Bild 3-1: Zustandsbasiertes Kollaborationsdiagramm der Systemanalyse für den Anwendungsfall „Pannemeldung durchführen“

Das „Steuergerät y am Komfort-Bus“ sendet einen Fehler an das Objekt vom Typ DiagnosticController. Der Initialzustand des Zustandsautomaten für den betrachteten Anwendungsfall ist „Ferndiagnose Off“. Kommt in diesem Zustand die Nachricht „IN 1“<sup>5</sup> (1: Fehler), so wird die objektinterne Aktion „aktiviereFehleranalyse“ gestartet und es findet ein Zustandsübergang in den Zustand „Fehleranalyse“ statt. Liefert die interne Aktion „aktiviereFehleranalyse“ das Ereignis „schwerwiegenderFehler“, so erfolgt ein Zustandsübergang in den Zustand „Verbindungsaufbau“ und ein Aussenden der Nachricht „OUT 2“ (2: Verbindungsaufbauanfrage). In diesem Zustand wird auf die Ankunft der Nachricht „IN 7“ (7: Bestätigung Verbindungsaufbau) gewartet. Beim Empfang der Nachricht „IN 7“ wird in den Zustand „Ferndiagnose On“ gewechselt und die Nachrichten „OUT 8“ (8: Fehler mit Steuergeräte-Adresse) und „OUT 10“ (10: Routingerlaubnis) werden versandt. Der Empfang der Nachricht 10 stellt wiederum ein Eingangsevent für den Zustandsautomaten des Objektes vom Typ RoutingController dar und bewirkt dort einen Zustandswechsel und das Senden der Nachricht OUT 11.

Für jeden Anwendungsfall sind nur die Zustände und Transitionen innerhalb eines Objektes sichtbar, die für diesen Ablauf relevant sind. Deshalb ist im Zustandsautomaten für das Objekt vom Typ DiagnosticController auch keine Transition vom Zustand „Ferndiagnose On“ in den Zustand „Ferndiagnose Off“ enthalten. Mit der Betrachtung aller relevanter Abläufe werden die Zustandsautomaten der Objekte schließlich vollständig.

<sup>5</sup> Man beachtete, dass die Nummern bei den Ereignissen und Aktionen der Zustandsautomaten der Nummerierung der Nachrichten im Kollaborationsdiagramm entspricht.

Die „Zustandsbasierten Kollaborationsdiagramme“ können sowohl in der Systemanalyse als auch im Systementwurf zur Beschreibung des dynamischen Verhaltens eingesetzt werden. In der Systemanalyse wird der Nachrichtenfluss zwischen den Objekten dargestellt und im Systementwurf werden konkrete Methoden von Objekten aufgerufen. Für die „Zustandsbasierten Kollaborationsdiagramme“ im Systementwurf bedeutet damit ein Ereignis IN x für einen Zustandsautomaten, dass die Methode mit der Nummer x aufgerufen worden ist. Eine Aktion OUT x bedeutet stets, dass die Methode mit der Nummer x des jeweiligen Objektes aufgerufen wird, zu dem der Pfeil des Kollaborationsdiagramms zeigt. Als Bedingung für eine Transition können die Rückgabewerte von Methodenaufrufen ausgewertet werden. Eine Notation [OUT x = OK] bedeutet dann, dass die Transition durchgeführt wird, wenn der Methodenaufruf mit der Nummer x den Rückgabewert OK liefert. Bild 3-2 zeigt ein „Zustandsbasiertes Kollaborationsdiagramm“ des Systementwurfs.

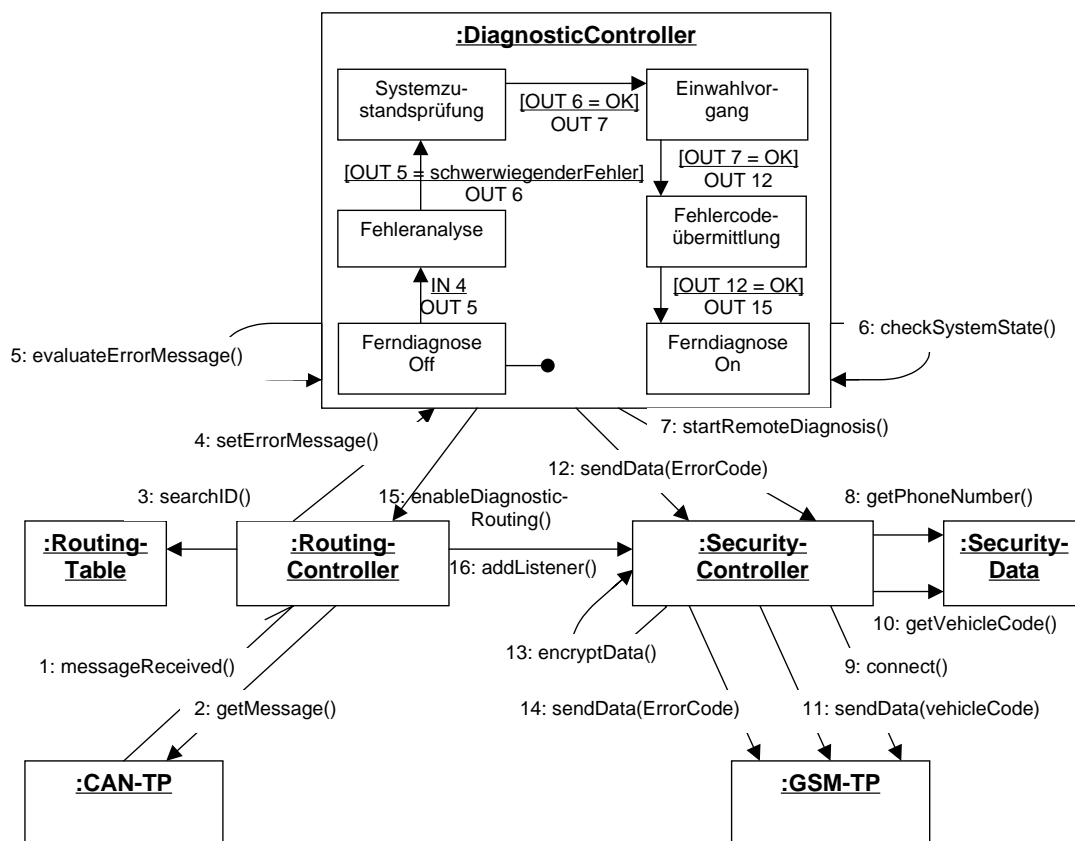


Bild 3-2 Zustandsbasiertes Kollaborationsdiagramm des Systementwurfs für den Anwendungsfall „Pannemeldung durchführen“

Methodenaufrufe, die in keinem Anwendungsfall einen Zustandswechsel eines Objektes bewirken, gehören zum transformatorischen Systemanteil und Methodenaufrufe, die einen Zustandswechsel bewirken, gehören zum reaktiven Systemanteil. Die Modellierung von transformatorischen und reaktiven Systemanteilen kann somit abgestimmt aufeinander erfolgen und die transformatorischen und reaktiven Systemanteile werden in den „Zustandsbasierten Kollaborationsdiagrammen“ deutlich sichtbar.

Mit einer entsprechenden Tool-Unterstützung können die Zustandsautomaten der Objekte in einem speziellen Anwendungsfall ein- und ausgeblendet werden und eine vollständige Darstellung eines einzelnen Zustandsautomaten für ein Objekt, der alle Zustände und Transitionen über alle Anwendungsfälle zeigt, ist entsprechend einfach möglich.

## 4 Zusammenfassung

Die „Zustandsbasierten Kollaborationsdiagramme“ stellen eine konsistente Modellierung der Dynamik in Systemanalyse und Systementwurf sicher. Dies ist mit einer getrennten Betrachtung von Zustands- und

Kollaborationsdiagramm nicht zu gewährleisten, da beide Diagramme unterschiedliche Sichten beschreiben und die Zusammenhänge nicht trivial ersichtlich sind. Eine vereinheitlichte Betrachtung bietet darüber hinaus den Vorteil, dass die Auswirkungen von Objektinteraktionen auf die Zustandsautomaten direkt sichtbar werden. Während bislang in einem Zustandsdiagramm nicht gesehen werden konnte, welche Zustände von welchen Anwendungsfällen beeinflusst werden, ist dies in den „Zustandsbasierten Kollaborationsdiagrammen“ auf den ersten Blick zu erkennen. Prägnant ausgedrückt kann gesagt werden, dass die Vereinheitlichung der Sichten in den „Zustandsbasierten Kollaborationsdiagrammen“ das Verständnis über die Funktionsweise des Systems erhöht und dass die Konsistenz der Systemmodellierung erheblich verbessert werden kann.

## 5 Literatur

- [1] Bruce Powel Douglass, „Doing Hard Time – Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns“, Addison-Wesley, 1999, ISBN: 0-201-49837-5
- [2] D. Rosenberg, K. Scott, „Applying Use Case Driven Object Modelling with UML“, 1. Auflage, Addison Wesley, ISBN: 0201730391, 2001
- [3] M. Fowler, „UML konzentriert – Strukturierte Einführung in die Standard-Objektmodellierungssprache“, 2. Auflage, Addison Wesley Longman Verlag GmbH, ISBN: 3827316170, 2000
- [4] G. Booch, J. Rumbaugh, I. Jakobson, „Das UML-Benutzerhandbuch“, 1. Auflage, Addison Wesley, ISBN: 3827314860, 1999
- [5] P. Hruschka, C. Rupp, „Agile Softwareentwicklung – für Embedded Real-Time Systems mit der UML“, 1. Auflage, Hanser-Verlag, ISBN: 3446219978, 2002