

# 1 Einleitung

## 1.1 Problematik

Zur Reduzierung der Produktionskosten werden in Unternehmen vielfach automatisierte Fertigungsanlagen eingesetzt. Durch hohe Investitionen in die Anlagen ist jedoch eine große zu produzierende Stückzahl erforderlich. Aufgrund einer steigenden Nachfrage der Konsumenten nach „individuellen“ Gütern sinkt indes die Losgröße und die Produktvarianten nehmen zu. Demnach müssen moderne Fertigungssysteme unproblematisch an veränderte Produktionsabläufe angepasst werden können.

Im Materialflussbereich werden zunehmend Transportsysteme entwickelt, die aus Modulen zusammengesetzt werden. Innerhalb dieser Systeme übernehmen vernetzte Steuerungseinheiten Überwachungs-, Steuerungs- und Regelaufgaben. Die Komplexität der Aufgaben erfordert die Verwendung effizienter Methoden und Werkzeuge bei der Entwicklung der Software [GKM00a]. In [KNN+00] wird ein Weg einer durchgängigen Spezifikation von der Analyse eines Produktionskontrollsystems bis zur Generierung des Codes der Steuerungssoftware aufgezeigt. Durch eine automatische Codegenerierung kann die Programmierung übersichtlich auf einer höheren Abstraktionsebene erfolgen. Zusätzlich verringern sich die Fehlerquellen, indem die Schnittstellen zwischen einzelnen Softwarekomponenten vom Spezifikationssystem konstant gehalten werden [GKM00a]. Obschon die Software unmittelbar aus der Spezifikation hervorgeht, ist vor ihrem Einsatz zu prüfen, ob sie den Anforderungen entspricht. Da Fehler in der Spezifikation im laufenden Betrieb des Systems zu hohen Kosten führen können, soll zuvor zur Validierung der Steuerung eine Simulation eingesetzt werden.

## 1.2 Zielsetzung

Ein Simulator ist ein Hilfsmittel, mit dem ein Simulationsmodell erzeugt und ausführbar gemacht werden kann. In dieser Arbeit soll ein Programmteil eines Simulators für dezentral gesteuerte schienengebundene Transportsysteme entwickelt werden, der eine Simulation der Hardware der Fördertechnik ermöglicht. Damit eine Validierung der Steuerungsspezifikation gewährleistet werden kann, ist eine Schnittstelle zur Kommunikation mit Objekten, die eine Steuerungssoftware simulieren, vorzusehen. Die erzeugten Daten sollen an die Benutzungsoberfläche weitergeleitet werden, um das Verhalten der virtuellen Anlage zu animieren.

---

In der vorliegenden Arbeit wird ausschließlich wegen der besseren Lesbarkeit nur die maskuline Form verwendet.

### 1.3 Vorgehen

Die Entwicklung des Simulators beruht auf einer objektorientierten Modellierung. Durch die Methoden objektorientierter Entwicklungsprozesse können die Phasen der Softwareentwicklung durchgängig aufeinander aufbauend ausgeführt werden. Aus der Gliederung dieser Arbeit sind die unterschiedlichen Phasen ersichtlich.

Im zweiten Kapitel wird eine grundlegende Einführung in die Terminologie der behandelten Bereiche gegeben. Es wird der Stand der Technik in Materialflusssystemen, Simulation und in der Notation objektorientierter Modelle aufgezeigt.

Aufbauend auf den zuvor ermittelten Grundlagen wird im dritten Kapitel zunächst die Anforderungsbeschreibung an den Simulator erstellt. Aus den dargestellten Forderungen ergibt sich eine Architektur, die den Simulator in die drei Teilbereiche Benutzungsoberfläche, Simulatorkern und Steuerungsebene aufgliedert. Die Entwicklung des Simulatorkerns steht im weiteren Verlauf im Vordergrund. Dieser Teilbereich erfordert eine Analyse der Komponenten und Prozesse schienengebundener Transportsysteme, um diese in Modellelementen nachzubilden. Diese Analyse sowie die Anforderungsanalyse der sich aus der Architektur ergebenden Schnittstellen beenden das Kapitel.

Während in der Analysephase die Klassen und ihre Beziehungen ermittelt werden, beziehen sich die Ausführungen des vierten Kapitels auf Aspekte des Designs. Zu Beginn des Kapitels wird die Wahl der verwendeten Simulationsmethode begründet. Anschließend wird die Umsetzung der Methode im Simulatorkern näher erläutert. Ferner wird die Modellierung der Schnittstellen zu den beiden anderen Architekturebenen diskutiert.

Um die Funktionsfähigkeit des entworfenen Modells zu belegen, wird der Kern im Anschluss an die Modellierungsphase implementiert. Auf Grundlage der Implementierung werden Testszenarien simuliert und das Laufzeitverhalten betrachtet. Im fünften Kapitel werden spezielle Aspekte der Implementierung sowie die Ergebnisse der Auswertungen dargelegt.

Eine Zusammenfassung der Arbeit erfolgt im sechsten Kapitel. Der Ausblick zeigt Erweiterungsmöglichkeiten des Simulators, wie die Anbindung einer interaktiven Virtual Reality Darstellung, auf.