

Bibliografische Beschreibung

Uwe Schob

Thema

Methode zur frühen virtuellen Inbetriebnahme von Steuerungsprogrammen durch halbautomatische Maschinenmodellbildung

Dissertation an der Fakultät für Maschinenbau der Technischen Universität Chemnitz, Institut für Werkzeugmaschinen und Umformtechnik IWU, Chemnitz, Januar 2012

+ 293 Seiten

+ 56 Abbildungen

+ 19 Tabellen

+ 127 Literaturverweise

Referat

Die Entwicklung von Maschinen und Anlagen ist ein zeitintensiver Prozess. Um ihn zu beschleunigen stellt diese Arbeit eine Methode vor, die auf der Wiederverwendung von Produktdaten beruht. Durch formale Regeln lassen sich Ausgangsdaten automatisch in eine andere Form, beispielsweise ein Simulationsmodell, übersetzen. Die Überprüfung eines Steuerungsprogrammes anhand einer virtueller Maschine ist dadurch mit nur geringem Modellerstellungsaufwand möglich.

Schlagworte

Anlagenentwicklung, Simulation, Modellerstellung, virtuelle Inbetriebnahme, Steuerungsprogrammtest, Datenwiederverwendung

Kurzfassung

Die vorliegende Arbeit beschäftigt sich mit dem Themenfeld der Entwicklung von Maschinen und Anlagen. Neben dem klassischen Vorgehen einer solchen Entwicklung werden aktuelle Optimierungsansätze, wie zum Beispiel die virtuelle Inbetriebnahme, vorgestellt. Trotz großen Potenzials finden die neueren Methoden nur schwer Einzug in die Praxis.

Diese Arbeit analysiert nun die Gründe für den Nicht-Einsatz und leitet daraus ein Anforderungsbild ab, welches im Wesentlichen aus der konsequenten Wiederverwendung von digitalen Produktdaten besteht. Mit der Hilfe von im klassischen Entwicklungsprozess bereits anfallenden Daten lässt sich der Aufwand für nachgelagerte Arbeiten erheblich reduzieren.

Am Beispiel der Erstellung von Simulationsmodellen für die virtuelle Inbetriebnahme von Steuerungsprogrammen wird eine Methode präsentiert, welche eine allgemeine Datenwiederverwendung ermöglicht. Die dazu notwendigen Voraussetzungen, der Ablauf sowie die erreichbaren Ergebnisse werden dargestellt.

In einer zweistufigen Umsetzungen wird die Methode verifiziert. Im ersten Schritt wird eine Rahmenanwendung realisiert, welche die Definition vielfältiger Transformationsprozesse ermöglicht. Im zweiten Schritt wurde ein konkreter Übersetzungsprozess um- und schließlich für ein Demonstrationsszenario eingesetzt. Das entstandene Werkzeug erlaubt die Übersetzung von Eingangsdaten in ein ablauffähiges Simulationsmodell.

Die Arbeit zeigt, dass eine formalisierbare Übersetzung verfügbarer Produktdaten die Entwicklung und Konstruktion beschleunigen kann. Da sich die Methode nahtlos in vorhandene Entwicklungsabläufe integrieren lässt und der Mehraufwand für eine Produktivitätssteigerung gering bleibt, ist ein ökonomischer Einsatz gegeben.

Abstract

The present work deals with the area of the development of machines and systems. In addition to the classic approach for development, current optimization techniques, e.g. virtual commissioning, are presented. Despite their large potential, they are hardly put into practise.

This work analyses the causes of the non-application and derives additional research requirements. They mainly consist of the consequent reuse of digital product data. By Usage of data created during the classic development process, the effort for downstream steps can be reduced considerably.

For the example of creating simulation models for the virtual commissioning of control programs, a method is presented, which allows the model generation out of reused / existing data. The therefor required preconditions, its workflow as well as the achievable results are presented.

The generic method is verified with a 2-step implementation. The first one includes the implementation of a framework, which allows the definition of various transformation processes. The second step represents the specialization of the former framework according for a specific use case and an exemplary system. The resulting application tool allows the transformation of several input documents into executable simulation models.

This work shows, that a formalizable transformation of existing product data can speed up the development process. As the method and its implementations can be integrated seamlessly into different development processes and application-scenes with minimal additional effort, an economical application is given.

Inhaltsverzeichnis

Abbildungsverzeichnis	13
Tabellenverzeichnis	17
Definitionsverzeichnis	19
Abkürzungsverzeichnis	21
1 Einleitung	23
1.1 Motivation	25
1.2 Zielstellung	26
1.3 Aufbau der Arbeit	28
2 Stand der Wissenschaft und Technik	31
2.1 Fertigungs- und Montageprozesse	31
2.1.1 Technische Prozesse	31
2.1.2 Prozessautomatisierung	32
2.2 Entwicklung technischer Systeme	34
2.2.1 Überblick	34
2.2.2 Mechanikkonstruktion	36
2.2.3 Elektrokonstruktion	40
2.2.4 Pneumatik- und Hydraulikkonstruktion	42
2.2.5 Steuerungstechnik	43
2.2.6 Forschungsarbeiten zur Verbesserung des Entwicklungsprozesses	46
2.3 Strukturelle methodische Erweiterungen	47
2.3.1 Mechatronik	47
2.3.2 Hochschulforschung	49
2.3.3 Virtuelle Produktentstehung	51

2.3.4	Integrierte Funktionsbeschreibung	54
2.3.5	Digitale Fabrik	56
2.3.6	Softwareentwicklung	58
2.4	Simulation als unterstützendes Werkzeug	59
2.4.1	Simulation von Montagesystemen	60
2.4.2	Simulation von NC-Maschinen	62
2.4.3	3D-Visualisierung von Anlagen	64
2.5	Kritik	66
3	Anforderungsprofil	69
3.1	Anwendungsszenario „Steuerungsprogrammtest“	69
3.2	Zielkriterien	73
3.2.1	Anforderungen bisheriger Ansätze	74
3.2.2	Industrielle Anforderungen	75
3.3	Zusammenfassung	76
4	Lösungsweg und Ansatz der AMG-Methode	79
4.1	Prinzip der AMG-Methode	79
4.2	Modellgenerierung und -übersetzung	83
4.2.1	Verfügbare Eingangsdaten	83
4.2.2	Modellzusammenführung	92
4.2.3	Wahl eines mechatronischen Metamodells	99
4.2.4	Modellübersetzung im Detail	105
4.3	Simulationsmodelle als Übersetzungsziel	110
4.3.1	Klärung des Einsatzzwecks	111
4.3.2	Auswahl einer Elementbibliothek	115
4.3.3	Kommunikation	118
4.4	Visualisierungsmodelle als Übersetzungsziel	123
4.4.1	Wichtige Modellaspekte	123
4.4.2	Aufbereitung von CAD-Daten	128
4.4.3	Darstellungsformate für 3D-Modelle	134
4.5	Zusammenfassung	137

5	Umsetzung der AMG-Methode	139
5.1	Methodenwerkzeug	139
5.1.1	Architektur	141
5.1.2	Entwicklungsstadien	144
5.1.3	Anwendung eines AMG-Projektes	148
5.2	Anwendungsszenario „Steuerungsprogrammtest“	150
5.2.1	Überblick des Szenarios	151
5.2.2	Eingangsdaten und Metamodell	152
5.2.3	Zieldaten	159
5.2.4	Formulierung der Transformationsregeln	164
5.2.5	Zusammenfassung	168
6	Erprobung der AMG-Methode	173
6.1	Beispielsystem	173
6.1.1	Statischer Aufbau	174
6.1.2	Dynamischer Ablauf	177
6.2	Ergebnisse	179
6.2.1	Transformation	179
6.2.2	Simulation	184
6.2.3	Visualisierung	186
6.2.4	Programmtest	187
6.3	Diskussion	190
6.3.1	Problemfelder des Beispiels	190
6.3.2	Beurteilung des AMG-Frameworks	192
6.3.3	Wirtschaftlichkeitsbetrachtung	193
7	Abschließende Betrachtung	197
7.1	Zusammenfassung	197
7.2	Schlußfolgerung	199
7.3	Ausblick	201
	Literaturverzeichnis	204
A	Aufbau mechatronisches Metamodell	221

B Bibliothek Automation	231
C Transformationsregeln im Detail	265