

Modularisierung SPS-gesteuerter mechatronischer Systeme

Jens Bathelt, Christian Bacs, Prof. Markus Meier

Zentrum für Produktentwicklung, ETH Zürich

Tannenstrasse 3, CH-8092 Zürich

Tel.: +41 1 632 35 15, Fax.: +41 1 632 11 81

E-mail: bathelt@imes.mavt.ethz.ch

Anders Jönsson

Department of Mechanical Engineering, Blekinge Institute of Technology

Campus Gräsvik, SE-371 79 Karlskrona

Tel.: +46 455 38 55 12, Fax.: +46 455 38 55 07

E-mail: anders.jonsson@bth.se

Zusammenfassung

Rein mechanische Lösungen werden zunehmend von mechatronischen Systemen abgelöst. Der X-by-wire Ansatz zeigt auch, dass der Anteil an mechanischen Teillösungen eines mechatronischen Systems immer weiter zurückgeht. Der vorliegende Beitrag skizziert einen Entwicklungsprozess für SPS-gesteuerte mechatronische Systeme, der dem immer wichtiger werdenden Steuerungsanteil Rechnung trägt. Dabei steht das parallele Arbeiten der Steuerungstechnik und der Konstruktion im Vordergrund, um die Entwicklungszeiten zu verkürzen und Fehler früher zu erkennen. Realisiert wird dies durch eine fachspezifische Modularisierung, deren erste Modulhierarchie als Ausgangslage der jeweiligen Disziplinen fungiert und somit einen parallelen Start ermöglicht. In dieser Arbeit werden Regeln vorgestellt, um die fachspezifischen Module von einer Funktionsstruktur abzuleiten. Abschliessend wird das Vorgehen anhand eines Industriebeispiels erläutert.

Schlüsselwörter

Mechatronik, Konzeptphase, Interdisziplinär, Concurrent Engineering, Produktentwicklung

1 Einleitung

Dieser Beitrag entstand im Rahmen des Projektes EVA (Early Virtual mAchine). Dessen Ziel ist es den Entwicklungsprozess SPS-gesteuerter mechatronischer Systeme zu optimieren. Speicher Programmierbare Steuerungen (SPS) zeichnen sich durch Ablaufprogrammierung, spezifiziert in der IEC 61131-3, aus [Bon97].

2 Mechatronisches Referenzsystem

In Abbildung 1 ist ein vereinfachtes mechatronisches System mit Stoff-, Energie- und Informationsflüssen [PB97] dargestellt. Die Versorgungsenergie wird hier nicht berücksichtigt.

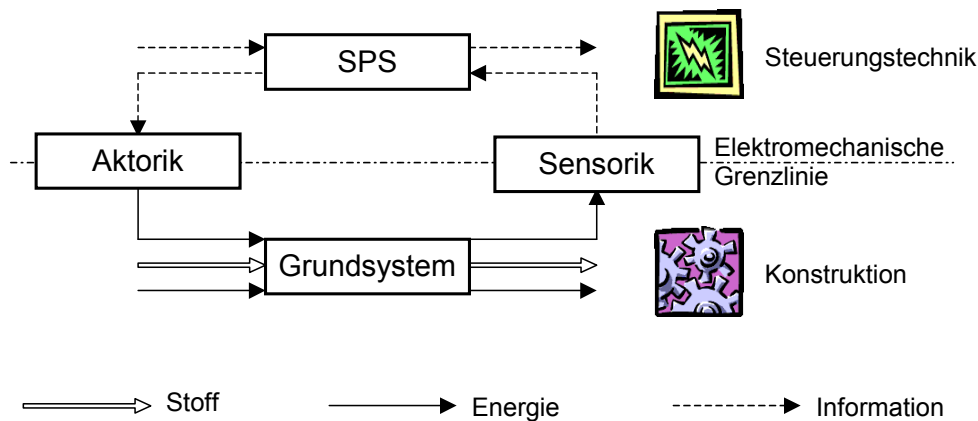


Abbildung 1: Mechatronisches Referenzsystem

Ähnliche Regelkreisdarstellungen mechatronischer Systeme finden sich auch in der Literatur [VDI03], [Gau00] und [Kal98], wobei in diesem Beitrag die informationsverarbeitende Komponente eine SPS ist.

3 Entwicklungsprozess

In der Abbildung 2 wird ein Weg zur Entwicklung SPS-gesteuerter mechatronischer Systeme aufgezeigt. Analog zur VDI 2221 [VDI93] wird ausgehend von der Anforderungsliste die Funktionsstruktur erstellt. Diese Funktionsstruktur enthält die klassischen Elemente wie Hierarchie, Sequenz und die Flüsse Energie, Stoff und Information [PB97], [Rot94], [Bre93]. Zusätzlich werden noch Transitionsbedingungen, wie sie z.B. in Petrinetzen vorkommen, verwendet [Bat03]. Dies erlaubt gerade im Hinblick auf die Ablaufsteuerungen eine vorteilhafte Modellierung der Funktionsstruktur [Buu90], [San01].

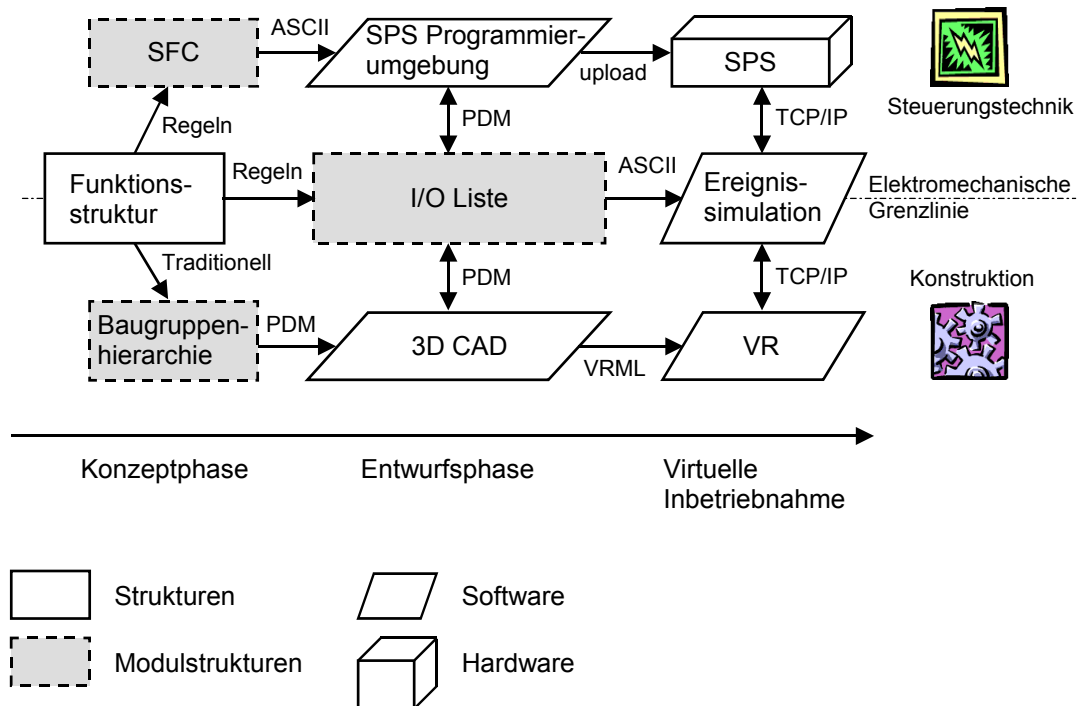


Abbildung 2: Vorgeschlagene Hauptkomponenten zur Entwicklung SPS-gesteuerter Systeme

Der kreative Prozess des Auffindens der Prinziplösungen in der Konzeptphase ist in der Abbildung 2 nicht berücksichtigt. Bei fortschreitender Modellierung der Funktionsstruktur sollen die Wirkprinzipien an die Funktionen „gehängt“ werden [Fre02], [PB97], [Gau02]. Die hier vorgestellte Modularisierung soll in erster Linie die interdisziplinäre Zusammenarbeit verbessern.

Heutzutage werden von der Baugruppenhierarchie (top-down) zuerst die Hauptbaugruppen z.B. im PDM definiert. Danach können mehrere Konstrukteure parallel diese Baugruppen weiter ausarbeiten. Dieser bewährte Ansatz wird nun für mechatronische Systeme ausgeweitet. Von der Funktionsstruktur wird nicht nur eine Baugruppenhierarchie [Mee02], sondern neu auch eine I/O Liste und eine Ablauflogik in SFC abgeleitet. Die resultierende Modulstruktur ist in Abbildung 3 dargestellt.

Analog zur Baugruppenhierarchie lassen sich mittels SFC auch Untermodule generieren, die von mehreren Steuerungsprogrammierern bearbeitet werden können. Dies ermöglicht ein paralleles Arbeiten der Steuerungstechniker und Konstrukteure, im Gegensatz zum (immer noch verbreiteten) sequenziellen Ansatz [Osm98], bei dem zuerst ein CAD-Modell erzeugt wird. Die Erfahrung hat allerdings gezeigt, dass es nicht möglich ist einer Baugruppe genau ein Steuerungsmodul zuzuordnen. SFC ist eine grafische Programmiersprache zur Analyse und Design von SPS Steuerungen [Bon97]. Sie ist eine von fünf kompatiblen Sprachen, die in der IEC 61131-3 normiert sind.

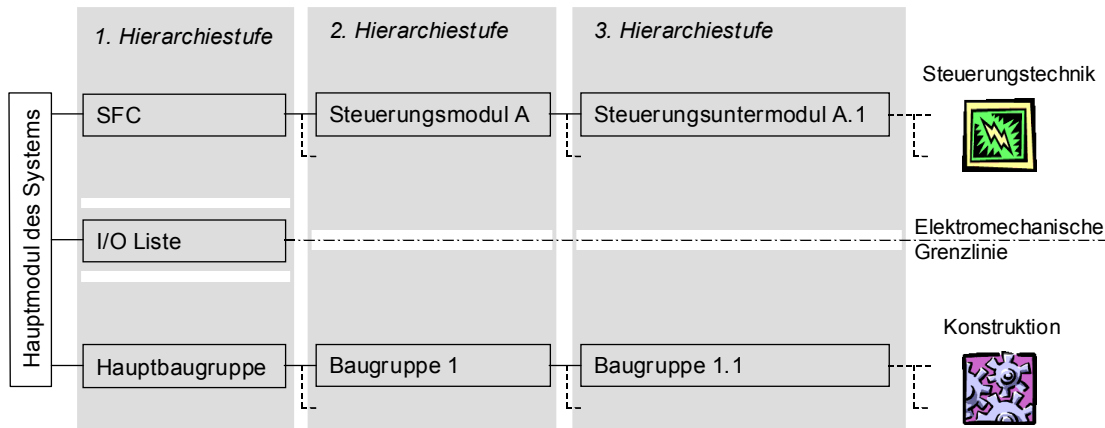


Abbildung 3: Vorgeschlagene Modulstruktur

Die I/O-Liste enthält die Daten zu den Aktoren (Outputs) und Sensoren (Inputs). Beide Disziplinen – Steuerungstechnik und Konstruktion – verändern und lesen diese Liste implizit oder explizit. Z.B. kann aus der Steuerungstechnik der Bedarf nach einem digitalen Input entstehen. D.h. ein Sensor mit einer zweiwertigen Logik, wie sie eine Lichtschranke bietet, wird benötigt. Diese Information wird beim vorgestellten Ansatz implizit über die Funktionsstruktur in die I/O-Liste einfließen. In diesem Beispiel kann der Konstrukteur einen passenden Sensor aus dem PDM ins CAD importieren, wobei diese Information über die Wahl des Sensors wiederum in die I/O-Liste fließt. Wie die Funktionsstruktur befindet sich auch die I/O-Liste auf der elektromechanischen Grenzlinie, da sie mechanische aber auch informationstechnische Daten beinhaltet. Im nächsten Kapitel wird beschrieben, wie sich die drei fachspezifischen Modulstrukturen ableiten lassen.

Da hier die Konzeptphase im Vordergrund steht, sollen die weiteren Schritte der Abbildung 2 nur noch kurz erläutert werden. Hat man die Ablauflogik in SFC abgeleitet, ist diese eine gute Grundlage, um in der jeweiligen SPS Programmierumgebung den Steuerungscode zu implementieren [Bon97]. Parallel dazu lässt sich die Geometrie im CAD definieren, nachdem die Baugruppenhierarchie im PDM erfasst ist.

Im Vorgängerprojekt ViMa (Virtual Machine) wurde die virtuelle Inbetriebnahme einer Maschine ermöglicht [Die02]. Dabei kommuniziert die SPS bidirektional über eine Ereignissimulation mit einem VR-Modell. Auch hierzu werden die Daten der I/O-Liste benötigt. Die kommerzielle Umsetzung der virtuellen Maschine ist über [Int03] erhältlich.

4 Ableitung der Modulstrukturen

In dieser Publikation steht der Weg von der Funktionsstruktur zu den fachspezifischen Modulen im Vordergrund. Die grafischen

Beschreibungssprachen der Funktions- und Modulstrukturen werden in [Bat03] vorgestellt.

4.1 Ableitung der Baugruppenhierarchie

Zur Ableitung der Baugruppenhierarchie gibt es schon viel Erfahrung und Publikationen [Mee02], [PB97], [Bre93] und [Fre02]. Auf diese bewährten Prozesse wird hier nicht näher eingegangen.

4.2 Ableitung der I/O-Liste

Eine Transitionsbedingung in der Funktionsstruktur weist auf einen Sensor hin. Die Funktionsstruktur eines automatischen Türöffners enthält z.B. die Funktion ‚Tür öffnen‘. Davor befindet sich die Transitionsbedingung ‚Person vor Tür‘. Erst wenn diese Bedingung erfüllt ist, wird die Funktion ‚Tür öffnen‘ aktiviert. Diese Information in der Funktionsstruktur führt hier sofort zu einem digitalen Input (Sensor) in der I/O-Liste.

Aus der Abbildung 1 wird ersichtlich, dass Aktoren neben der Versorgungsenergie ein Informationssignal erhalten und Energie an das Grundsystem weiter geben. Im Beispiel des Türöffners wird die Steuerung einen Befehl zum Aktivieren an den Aktor senden, nachdem der Sensor eine Person detektiert hat. Der Aktor wandelt nun mit der Information die Versorgungsenergie in mechanische Energie um. Diese mechanische Energie wird nun die Tür öffnen. D.h. Funktionen, die einen Informations- und Energiefluss als Eingang und einen Energiefluss als Ausgang haben, beschreiben einen Aktor.

Am Anfang wird in der I/O-Liste lediglich der Bedarf an geeigneter Aktorik und Sensorik dokumentiert sein (vgl. Tabelle 1). Der Signalname (Person vor Tür), der Typ (Sensor oder Aktor) und das Datenformat wie digital (Lichtschranke) oder analog (Temperatursensor) ist bekannt. Dies reicht schon vollständig aus, um die Steuerung zu programmieren. Für die Simulation des Maschinenverhaltens in der virtuellen Inbetriebnahme benötigt man die Spezifikation der Aktorik/Sensorik. Die gewählten Wirkprinzipien werden die Wahl der Aktoren/Sensoren beeinflussen. Oft sind die verfügbaren Aktoren und Sensoren, wie Normteile, schon im PDM System erfasst und deren Geometrie kann ins CAD importiert werden. Neu wird zusätzlich der ausgewählte Aktor oder Sensor in der entsprechenden Zeile der I/O Liste eingetragen.

Beim Erstellen der Funktionsstruktur stellt sich auch immer die Frage, wie viele Hierarchiestufen, also wie ‚tief‘, man modellieren soll. Hat man die Stufe erreicht, in der man Aktoren und Sensoren identifizieren kann, so ist dies ein Indiz dafür, dass man keine weiteren Hierarchiestufen benötigt. Dies ergibt sich aus der

Tatsache, dass die Aktoren und Sensoren bei der Herstellung von SPS-gesteuerten Maschinen in der Regel über einen Zulieferer bezogen werden.

4.3 Ableitung der Ablauflogik in SFC

In dieser Arbeit wird vorgeschlagen, die Ablauflogik in SFC zu beschreiben. Diese Sprache wird zum Einen in der Konzeptionsphase der SPS-Programmierung eingesetzt [Bon97], zum Anderen ist SFC formal nah an der Funktionsstruktur, da sie auch Hierarchie, Sequenz und Transitionsbedingungen beinhaltet. Dieser Zusammenhang ist aber nicht nur formaler Natur, sondern es ergeben sich auch inhaltlich Parallelen. Der Prozessablauf ist in der Funktionsstruktur enthalten, wenn nicht nur ein hierarchisches Funktionsmodell, sondern auch ein sequenzielles Modell erstellt wird [Bre93]. Dieser Prozess wird durch die Ablauflogik im SFC beschrieben.

Um die Ablauflogik aus der Funktionsstruktur herauszulösen, kann wieder das mechatronische Referenzsystem in Abbildung 1 zu Hilfe genommen werden. Interessant für die Steuerung sind alle Funktionen, die über den Informationsfluss mit der SPS verbunden sind. Anders ausgedrückt: Funktionen innerhalb der Grundstruktur, die ausschliesslich Energie- oder Stofffluss als Ein- oder Ausgang haben, werden nicht in die Ablaufstruktur für die Steuerung übernommen.

4.4 Zusammenfassung der Regeln zur Ableitung der Modulstrukturen

Mit der oben beschriebenen Methodik ergeben sich verschiedene Ausformulierungen der drei Regeln. Im Folgenden werden diese kurz zusammengefasst:

- **Sensordetektion:**
Eine Transitionsbedingung impliziert einen Sensor.
- **Aktordetektion:**
Eine Funktion, dessen Eingangsflüsse aus Versorgungsenergie und Information bestehen und die einen Energiefluss an das Grundsystem weitergibt, indiziert einen Aktor.
- **Ableitung der Ablauflogik:**
Es werden alle Funktionen ausgeblendet, die ausschliesslich Energie- oder Stofffluss als Ein- oder Ausgang haben.

5 Beispiel

Anhand des automatischen Fließwechslers einer Kämmereimaschine Abbildung 4 werden hier die Regeln aus dem Absatz 4 exemplarisch angewandt. Acht Fließrollen werden hinten zugeführt und der Fließ wird parallel gekämmt.



Abbildung 4: Kämmereimaschine

Früher mussten die Rollen manuell gewechselt werden, sobald eine Rolle leer war. Im Rahmen einer partiellen Neuentwicklung soll dies nun vollautomatisch geschehen. Dazu benötigt man einen automatischen Fließwechsler. Dessen Hauptfunktion ist in Abbildung 5 dargestellt.

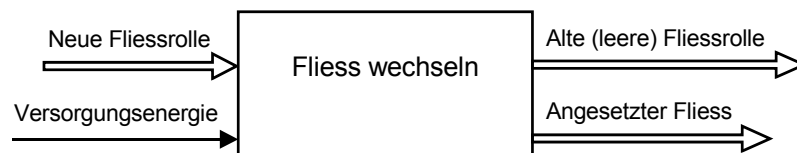


Abbildung 5: Hauptfunktion des Fließwechslers

Die erste Hierarchiestufe der Funktionsstruktur wird in der Abbildung 6 gezeigt. Die Darstellung lehnt sich an die SFC Normierung an. Alle klassischen Elemente sind in der Abbildung 6 enthalten. Darüber hinaus findet sich hier eine Transitionsbedingung ‚Fließ ausgegangen‘. Der automatische Fließwechsel darf erst dann starten, wenn die aktuelle Rolle leer ist.

Analog zu den Petrinetzen oder eben SFC wird diese Funktion als kleiner Balken angedeutet. Erst wenn diese Bedingung erfüllt ist, kann der Prozess fortschreiten. Wie in 4.2 besprochen, weist dies sofort auf einen Sensor hin. Damit ergibt sich jetzt schon ein erster Eintrag in der I/O Liste in der Tabelle 1. Aus Gründen der Übersichtlichkeit sind nicht alle Transitionsbedingungen und Flüsse in der Abbildung 6 aufgeführt.

Die Aktoren findet man typischerweise in der untersten Hierarchiestufe der Funktionsstruktur. Modelliert man von der Funktion ‚Alte Rolle entfernen‘ jeweils rekursiv immer die erste Unterfunktion, erhält man die Funktionsstruktur der Abbildung 7.

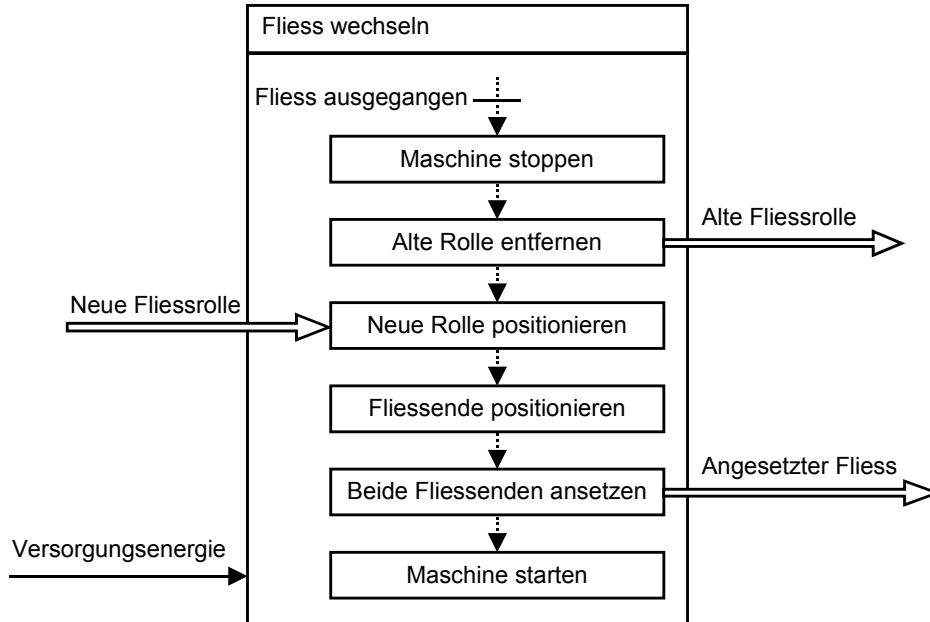


Abbildung 6: Erste Hierarchiestufe der Funktionsstruktur des Fließwechslers

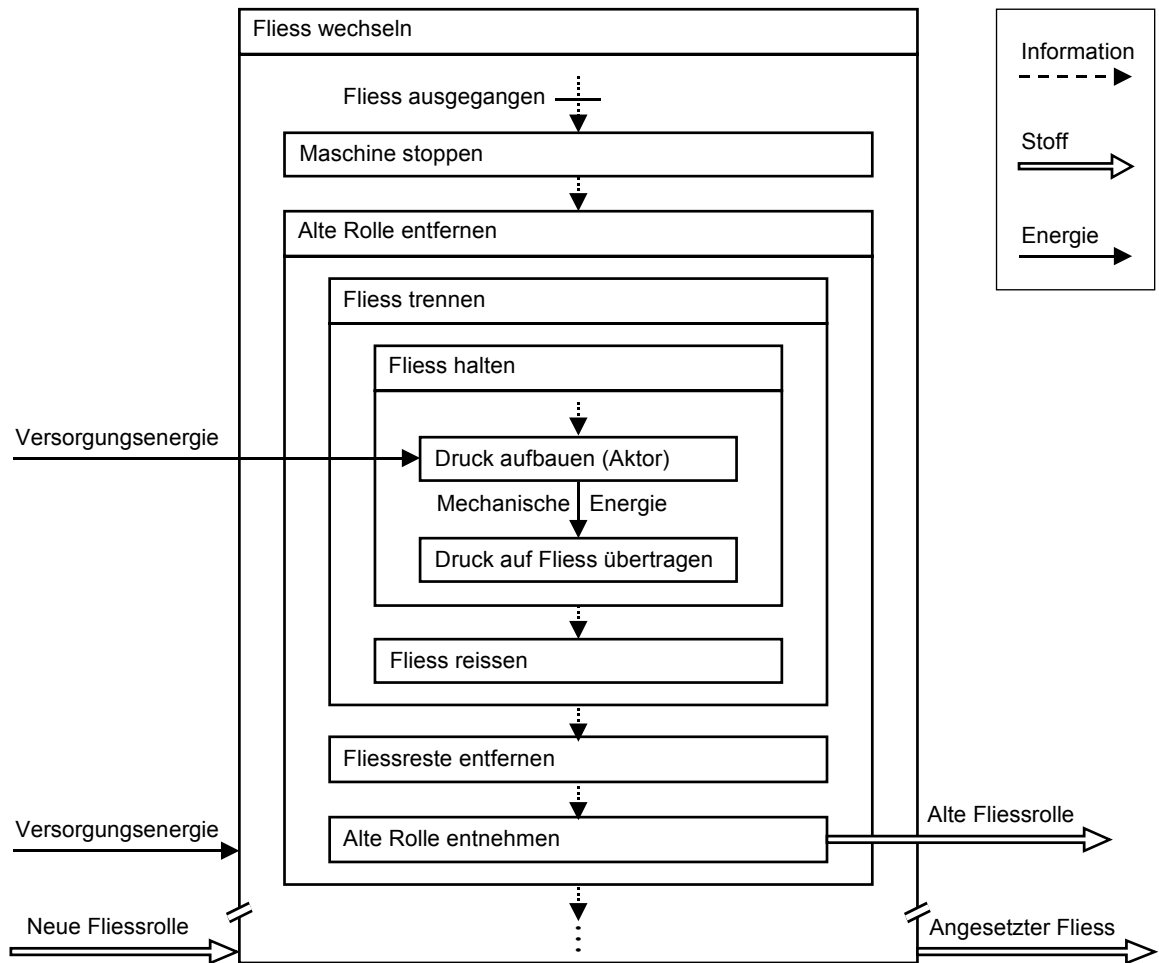


Abbildung 7 Funktionsstruktur mit Aktor

Die Funktion ‚Fließ halten‘ enthält einen Aktor. Dieser zeichnet sich dadurch aus, dass – wie in Abschnitt 4.2 beschrieben – die Funktion neben der Versorgungsenergie einen Informationsfluss als Eingang und einen Energiefluss als Ausgang hat. Dieses Wissen wird wiederum in die I/O Liste eingetragen (Tabelle 1).

Tabelle 1: Erste Version der I/O Liste

Id	Typ	Grösse	Name	PDM Referenz	Bemerkung
1	Input	Digital	Fließ ausgegangen		
2	Output	Digital	Druck aufbauen		Fließ halten

Wie in Abschnitt 4.3 dargelegt, interessiert der rein mechanische Teil des Prozesses die Steuerung nicht. Um das Grundgerüst für das Ablaufprogramm in SFC zu erhalten, muss nun dieser mechanische Anteil unterdrückt werden. In der Funktion ‚Fließ halten‘ wäre dies die Funktion ‚Druck auf Fließ übertragen‘. Die nächste wesentliche Funktion für die Steuerung wäre ‚Fließ reißen‘. Diese

enthält wieder ein Aktorsignal. Somit erhält man durch das Herauslösen aller Aktor- und Sensorfunktionen den steuerungsrelevanten Anteil. Dieser ist nun schon inhärent mittels SFC beschrieben.

Somit erhält man das Untermodul für die Steuerungsprogrammierer. Dieses Untermodul lässt sich wiederum – über die erste Hierarchiestufe der Funktionsstruktur (Abbildung 6) – in die vier Untermodule 'Alte Rolle entfernen', 'Neue Rolle positionieren', 'Fließende positionieren' und 'Beide Fließenden ansetzen' aufteilen. Diese vier Untermodule repräsentieren nun die 2. Hierarchiestufe des steuerungsrelevanten Anteiles der Modulstruktur (Abbildung 3) und können nun parallel in der Steuerungstechnik bearbeitet werden.

6 Ausblick

Das vorgestellte neue Konzept zum parallelen Arbeiten der Steuerungstechnik und Mechanik wurde in der Modularisierung vertieft. Das weitere Vorgehen wird Gegenstand weiterer Arbeiten und Veröffentlichungen werden. Dies betrifft das Verwenden der Module in der SPS Programmierumgebung und im CAD. Auch die Anbindung und Nutzung des PDM wurde in diesem Beitrag lediglich gestreift. Nach der Entwurfsphase wird die virtuelle Inbetriebnahme durch eine Anbindung an die Virtuelle Maschine unterstützt.

Buur weist in [Buu89] auf die Notwendigkeit von domänenübergreifenden Beschreibungssprachen hin, welche dem Entwickler eine lösungsneutrale und funktionale Beschreibung ermöglichen. Die Beschreibungssprachen für die hier vorgestellten Funktions- und Modulstrukturen werden in [Bat03] ausführlich dargelegt.

Literatur

- [Bat03] Bathelt, J.; Jönsson, A.; Bacs, C.; Kunz, A; Meier, M.: Conceptual design approach for mechatronic systems controlled by a programmable logic controller (PLC), ICED03, Schweden, 2003.
- [Buu89] Buur, J.; Andreasen, M.: Design models in mechatronic product development, Journal of Design Studies Vol.10 Nr.3, 1989.
- [Buu90] Buur, J.: A Theoretical Approach to Mechatronics Design, Technical University of Denmark, Dissertation, 1990.
- [Bon97] Bonfatti F., Monari P. D. and Sampieri U.: IEC 1131-3 Programming Methodology, CJ International, France, 1997.
- [Bre93] Breiing A.; Flemming M.: Theorie und Methoden des Konstruierens, Springer-Verlag Berlin Heidelberg New York, 1993.

- [Die02] Dierssen S.: Systemkopplung zur komponentenorientierten Simulation digitaler Produkte, Fortschr.-Ber. VDI Reihe 20 Nr. 358 VDI Verlag, Düsseldorf, 2002.
- [Fre02] Frei, N.: Der Konstruktionsprozess: Wirkmodelle basierend auf Constraints. Dissertation, ETH Zürich, Nr.14533, 2002.
- [Gau00] Gausemeier, J.; Lückel, J.: Entwicklungsumgebungen Mechatronik: Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme, HNI-Verlagsschriftenreihe Bd.80, Paderborn, 2000.
- [Gau02] Gausemeier, J.; Köckerling, M.: Systematische rechnerunterstützte Verträglichkeitsuntersuchung in der Konzeptphase mechatronischer Systeme, 13. Symposium „Design for X“, 2002.
- [Int03] Intelliact AG: <http://www.intelliact.ch/>, 2003.
- [Kal98] Kallmeyer, F.: Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme, HNI-Verlagsschriftenreihe, Paderborn, 1998.
- [Mee02] Koch, M.; Meerkamm, H.: Durchgängige Funktionsmodellierung in den frühen Konstruktionsphasen, 13. Symposium „Design for X“, 2002.
- [Osm98] Osmers, U.: Projektieren speicherprogrammierbarer Steuerungen mit Virtual Reality, Karlsruhe, Institut für Werkzeugmaschinen und Betriebstechnik der Universität Karlsruhe, Dissertation 1998.
- [PB97] Pahl, G.; Beitz, W.: Konstruktionslehre - Methoden und Anwendung. 4. Aufl., Springer-Verlag Berlin Heidelberg, 1997.
- [San01] Santos A. P.; Cury J.E.R.; De Negri V. J.: Mechatronic Design and Supervisory Control Theory for Fluid Power Applications, 5th International Conference on Fluid Power Transmission and Control, Hangzhou, China, 2001.
- [VDI03] N.N.: VDI-Richtlinie 2206 Entwicklungsmethodik für mechatronische Systeme. VDI-Verlag, Düsseldorf, liegt voraussichtlich im Februar 2003 als Gründruck vor.
- [VDI93] N.N.: VDI-Richtlinie 2221 Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, VDI-Verlag, Düsseldorf, 1993.