

# 1 Einleitung

## *Introduction*

Produktionstechnische Systeme zur automatisierten Herstellung von Wirtschafts- und Konsumgütern, wie pharmazeutische Waren, Holz- oder kunststoffliche Teile, elektronische Geräte, Lebensmittel oder Metallerzeugnisse, sind sowohl in ihren Ausprägungen wie Anwendungsbereichen durch eine starke Diversität geprägt.

Die Architekturvarianz der produktionstechnischen Systeme steht dabei in direkter Analogie zur Vielfalt der produzierten Güter. Maßgeblich strukturbestimmend sind hierbei, insbesondere im Falle formgebender Maschinen, die (modulübergreifend) zu implementierenden Wirk- und Interaktionsprinzipien mechatronischer Module, um die zur *Umsetzung des Fertigungsprozesses erforderlichen Anlagenfähigkeiten* zu realisieren. Da ein Prozessschritt zumeist eine spezifische Kombination mehrerer Basisprozesse (Bewegungsführung, füge-technische oder thermodynamische Prozessführung etc.) erfordert, sind diese zumeist als stark interdisziplinär geprägte mechatronische Funktionseinheiten ausgeführt, wie es die typische Prozesskette einer Flaschenabfüllung<sup>1</sup> verdeutlicht.

Derartig variante Systemarchitekturen, stetig steigende Funktionsdichten mechatronischer Baugruppen sowie der Trend zu verkürzten Produktlebenszyklen bilden jedoch ein bedeutendes Spannungsfeld für den Produktentwicklungsprozess produktionstechnischer Systeme [ABRA16; GAUS13; BREC12]. So müssen aufgrund der hohen disziplinübergreifenden Abhängigkeiten u. a. funktionale Schnittstellen, parametrische Prozesskonfigurationen sowie geometrische Gegebenheiten untereinander kontinuierlich ausgetauscht und aktualisiert werden, da potentiell jede Systemänderung wesentliche Auswirkungen auf andere Disziplinen haben kann [HEIH16].

Gleichzeitig ist die methodische Kollaboration im Entwicklungsprozess vieler Maschinen- und Anlagenbauern durch dezentrale Systemspezifikationen und heterogene Kommunikationsstrukturen geprägt, die u. a. zu hohen kommunikativen Aufwänden, Reibungsverlusten und bedeutenden Erstellungsaufwänden informaler Austauschdokumente führen.

Die starke Zunahme der erforderlichen interdisziplinären und datendurchgängigen Kooperation wird daher zunehmend als elementare Herausforderung innerhalb des Produktentwicklungsprozesses wahrgenommen [GAUS13], da die verspätete Identifikation fehlerhafter Designentscheidungen hohe Änderungskosten nach sich ziehen [INCO15].

Eine wirtschaftlich effiziente Entwicklung und Konstruktion derartig hochvarianter und domänenübergreifender Abhängigkeiten funktionaler Einheiten erfordert daher eine datenkonsistente, interdisziplinäre sowie weiterverwendbare Architektur- und Schnittstellenspezifikation.

---

<sup>1</sup>Reinigung, Transport und Schadensinspektion der Flasche, Abfüllung, Etikettierung, Bündelung, Sekundärverpackung und Palettierung

Einen geeigneten methodischen Ansatz bietet hierzu das Model-Based Systems Engineering (MBSE), welches ursprünglich für verifikationsintensive Raumfahrtprojekte entwickelt wurde [INCO14]. Basierend auf einer expressiven Sprachsyntax und Semantik erarbeitet das interdisziplinäre Team ein anforderungsorientiertes Lösungskonzept als Systemmodell. Der kollaborative Entstehungsprozess prägt dabei ein gemeinsames Systemverständnis aus, wobei das Modell ferner als *zentrales* Architektur- und Datenmodell im Entwicklungsprozess dienen kann.

Die disziplinübergreifend formulierten technischen Lösungskonzepte können unter Berücksichtigung systemischer und funktionaler Abhängigkeiten analysiert und anschließend iterativ in das Modell integriert werden. Anforderungsreferenzen dokumentieren dabei den funktional korrekten Entwurf. Die so sukzessiv entstehende Wissensbasis kann dann in der domänenspezifischen Entwurfsphase zur Extraktion von Designentscheidung und Generierung von Partialmodellen genutzt werden.

Für die modellbasierte Spezifikation mechatronischer Systeme liegen bereits sowohl generalistische (GPML) wie domänenspezifische (DSML) Sprachen vor, die sich zwar in der Entwicklung eingebetteter Systeme und Software etablieren, jedoch nicht innerhalb der produktionstechnischen Domäne [BREC16c; HABE16; MUNK16; GÖLZ15]. Dies ist sowohl dem im Kontrast zum maschinenbaulich geprägten Anlagenbau stehenden abstrakten, Informatik-orientierten Typ-Instanz-Sprachparadigma, sowie der teilweise unzureichenden Berücksichtigung wesentlicher produktionstechnischer Systemeigenschaften in den Sprachnotationen geschuldet. Ferner hemmen Medienbrüche zwischen den disziplinspezifischen Entwicklungswerkzeugen die Integration von Architekturmodellen [FLEN19; FOEH17; CA-DE17a; GEIS16].

Das Ziel der vorliegenden Arbeit ist es daher, eine dediziert für die produktionstechnische Domäne angepasste Spezifikationssprache zu entwickeln, die eine interdisziplinäre und intuitive Systemmodellierung phasenübergreifend von der Anforderungsspezifikation bis in die domänenspezifische Entwicklung unterstützt. Ein modellkonsistenter, *funktionaler* Modellierungsansatz soll die Bildung eines übergreifenden Systemverständnisses, der kreativen Lösungsfindung, sowie der Wieder- und Weiterverwendung von Templates und Altprojekten fördern. Das Architekturmodell soll zudem auf Basis eines datendurchgängigen Konzepts die Integration und Generierung domänenspezifischer Partialmodelle als zentrales Datenmodell ermöglichen.

Der Ansatz einer domänenspezifischen Spezifikationssprache ist damit ein Beitrag zur Effizienzoptimierung der produktionstechnischen Systementwicklung durch die Reduzierung des Abstimmungs- und Dokumentationsaufwand mittels einer semantischen Notation und der Formulierung eines elementspezifischen funktionalen Implementierungskontextes.

# Introduction

Production systems for the automated manufacturing of economic and consumer products, such as pharmaceutical goods, wooden or plastic parts, electronic devices, food and beverages or metal products, are characterized by a high degree of diversity, both in terms of their characteristics and areas of application.

The architectural variance of production technology systems is directly analogous to the diversity of the goods produced.

The decisive structural factors here, especially in the case of form-giving machines, are the principles of operation and interaction of the to be implemented mechatronic assemblies (across modules) in order *to realize the system capabilities required for implementation of the production process*. Since a process step usually requires a specific combination of several basic processes (motion control, joining technology or thermodynamic process control etc.), these are usually designed as strongly interdisciplinary mechatronic functional units, i.e. in a typical process chain<sup>2</sup> of a bottle filling plant.

However, such variant system architectures, constantly increasing function densities of mechatronic modules and the trend towards shorter product lifecycles form a significant area of conflict for the product development process of production engineering systems [AB-RA16; GAUS13; BREC12]. Due to the high level of interdependencies between disciplines functional interfaces, parametric process configurations and geometric conditions must be continuously exchanged and updated, since every system change can potentially have significant effects on other disciplines [HEIH16].

At the same time, the methodical collaboration in the development process of many mechanical and plant engineering companies is characterized by decentralized system specifications and heterogeneous communication structures, which, among other things, lead to high communicative efforts, friction losses and significant efforts to create informal exchange documents.

The strong rise of the required interdisciplinary and data-consistent cooperation is therefore increasingly perceived as a fundamental challenge within the product development process [GAUS13], since the late identification of faulty design decisions results in high modification costs [INCO15].

An economically efficient development of such highly variant and cross-domain dependencies of functional units therefore requires data consistent, interdisciplinary and reusable architecture and interface specifications.

A suitable methodological approach is offered by MBSE, which was originally developed

---

<sup>2</sup>Cleaning, transport and damage inspection of the bottle, filling, labelling, bundling, secondary packaging and palletising

for verification-intensive (aero)space projects [INCO14]. Based on an expressive language syntax and semantics, the interdisciplinary team develops a requirements-oriented solution concept as a system model. The collaborative development process is characterized by a common understanding of the system, whereby the system model can also serve as a *central* architecture and data model in the development process.

The discipline-spanning developed technical solution concepts can be analyzed under consideration of systemic and functional dependencies and subsequently be iteratively integrated into the system model. Requirement references specify the functional correct design. This incrementally created knowledge base can then be used in the domain-specific design phase to extract design decisions and to generate partial models.

For the model-based specification of mechatronic systems there are already generalistic (GPML) and domain-specific (DSML) languages available. These languages are established in the development of embedded systems and software, but not within the production domain [BREC16c; HABE16; MUNK16; GÖLZ15]. This is due both to the abstract, computer science-based type-instance language paradigm, which contrasts with the mechanical engineering dominated plant engineering, and to the partly insufficient consideration of essential production-related system properties in the language notations. Furthermore, media breaks between the discipline-specific development tools inhibit the integration of architecture models [FLEN19; FOEH17; CADE17a; GEIS16].

The goal of this thesis is therefore to develop a specification language that is specifically adapted to the production domain and supports interdisciplinary and intuitive system modeling across all phases, from requirements specification to domain-specific development. A model-consistent, *functional* modelling approach should promote the establishment of a comprehensive system understanding, creative solution finding, as well as the reuse and further application of templates and former projects. The architectural model should also enable the integration and generation of domain-specific partial models as a central data model on the basis of a data-consistent concept.

The approach of a domain-specific specification language is thus a contribution to the efficiency optimization of production system development by reducing the coordination and documentation effort by means of a semantic notation and the formulation of an element-specific functional implementation context.