

Physische Architekturen variantengerecht aus Funktionalen Architekturen für Systeme (FAS) spezifizieren

Sylvia Melzer, Ulrike Wittke, Hartmut Hintze, Ralf God

Technische Universität Hamburg-Harburg, Institut für Flugzeug-Kabinensysteme,
Nesspiel 5, 21129 Hamburg, {sylvia.melzer, ulrike.wittke, hartmut.hintze,
ralf.god}@tuhh.de

Zusammenfassung: Mit der wachsenden Funktionalität von Systemen und mit zunehmenden Systemvarianten steigt gleichzeitig auch die Komplexität. Den sich daraus ergebenden Herausforderungen, insbesondere der Konsistenthaltung der Systementwicklung zu den Anforderungen über alle Entwicklungsstufen hinweg, wird heute mit dem Ansatz des Model-Based Systems Engineering (MBSE) begegnet. Es muss dabei möglich sein, verschiedene Systemvarianten im Modell zu spezifizieren, ohne für jede Variante ein Duplikat des Basismodells erstellen zu müssen. Vielmehr sollte eine variantengerechte Strukturierung auf Basis von Funktionen erfolgen, und es sollten lediglich die varianten Anteile in den physischen Architekturen abgebildet werden. Dieser Beitrag zeigt, wie die bekannte FAS-Methode (FAS: Funktionale Architekturen für Systeme) gewinnbringend für die modellbasierte und variantengerechte Spezifikation physischer System-Architekturen genutzt werden kann.

1 Einleitung

Die wachsende Funktionalität von Systemen, welche fast immer mit unterschiedlichen Systemvarianten einhergeht, führt zu steigender Komplexität bei den Systemen und beim Systementwurf. Ein Lösungsansatz, um den daraus erwachsenden Herausforderungen bei der Entwicklung komplexer Systeme zu begegnen, ist der Ansatz des Model-Based Systems Engineering (MBSE). Das Erfordernis, im Modell verschiedene Systemvarianten abbilden zu können, gewinnt dabei immer mehr an Bedeutung, da die Industrie über ein möglichst umfängliches Basis-Modell des Systems eine kostensparende Standardisierung erreichen möchte und dennoch über die Varianten eine Individualisierung gegenüber den Kunden zulassen will. Eine Systemvariante ist durch ein Basis-Modell und durch eine jeweilige Differenz charakterisiert. Das Basis-Modell repräsentiert den Systemkern und die Differenz den jeweiligen Unterschied von Systemkomponenten pro Variante [Me14]. Hinsichtlich eines Prozessmodells zeichnet sich beispielsweise die Luftfracht-Transportkette durch verschiedene Varianten von Prozessen aus. Der Versand von Fracht innerhalb Deutschlands und der Versand weltweit stellen hierbei zwei typische Varianten dar. Im Forschungsprojekt SiLuFra (Sichere Luftfracht-Transportkette) [Si13] sollten die Sicherheit und die Effizienz der Transportprozesse durch neue Prozessvarianten verbessert werden. Dabei lag das Hauptaugenmerk im Projekt nicht auf der Entwicklung neuer technischer Lösungen,

Copyright © 2016 Sylvia Melzer. Zur Veröffentlichung und Nutzung durch GfSE und die mit ihr verbundenen Organisationen freigegeben.

sondern es galt, die bestehenden Prozesse mittels Varianten zu optimieren und potentiell um weitere prozess- und organisationsgeleitete Maßnahmen zu ergänzen. Ein im Projekt erarbeiteter und optimierter Prozess wird als Trusted-Forwarder-Konzept bezeichnet.

Passende Methoden und Werkzeuge des Model-Based Systems Engineering (MBSE) erfordern in diesem Zusammenhang,

- dass aus den Anforderungen eine funktionale Architektur für Systeme abgeleitet werden kann, um eine Optimierung anhand von Prozessvarianten vornehmen zu können. In diesem Kontext setzt sich eine funktionale Architektur aus verschiedenen Funktionen sowie Akteuren zusammen und beschreibt deren logisches Zusammenwirken. Eine funktionale Architektur liefert damit Informationen über Wirkungszusammenhänge, die über die Beschreibung einzelner Funktionen hinausgehen.
- dass Varianten auf Basis der funktionalen Architektur spezifiziert werden können und
- dass damit der Lösungsraum möglicher physischer Architekturen eingegrenzt werden kann. Die physische Architektur setzt sich aus verschiedenen Hardwarekomponenten zusammen und beschreibt deren physische Verbindungen. Die physische Architektur liefert Informationen über die Anzahl von Systemkomponenten, über physische Schnittstellen und gegebenenfalls auch Einbauinformationen.

Um diese Erfordernisse erfüllen zu können, gilt es einen Weg zu finden, welcher bei der Herleitung einer physischen Architektur für Systeme

- ein geeignetes prozessuales Vorgehen definiert,
- Methoden zur Strukturierung von Varianten nutzt,
- eine Funktionale Architektur für Systeme (FAS) als Basis nutzt und
- von Softwarewerkzeugen unterstützt wird.

Schließlich muss dieser Weg dazu beitragen, dass

- die im Problemraum definierte funktionale Architektur dazu genutzt werden kann, die Anzahl physischer Architekturen im Lösungsraum einzugrenzen,
- alle Komponenten eines Systems gezielt entwickelt werden können und
- die Systemanforderungen nachverfolgt und abgeleitete Anforderungen bzw. Verfeinerungen hinsichtlich ihrer Entstehung untersucht werden können.

In diesem Beitrag wird gezeigt, wie sich ein solcher Weg mit einer Kombination von Methoden für die modellbasierte Systementwicklung unter Verwendung der semi-formalen Modellierungssprache *Systems Modeling Language* (SysML) nach Holt *et al.* [HPB12], nach Weilkiens [We16] und nach Lamm und Weilkiens [LW15] zur Herleitung Funktionaler Architekturen für Systeme (FAS-Methode) für die variantengerechte Spezifikation physischer System-Architekturen realisieren lässt.

2 Verwendete Methoden

Modellbasierter Systementwicklungsprozess: In der ersten Phase des Systems-Engineerings, dem *Requirements Engineering*, werden die Sichtweisen aller Stakeholder, vgl. z.B. [KG15], berücksichtigt und analysiert. Holt *et al.* definieren in [HPB12] ein Vorgehen die Anforderungen aller Stakeholder zu erheben und während der gesamten Systementwicklung verwalten zu können. Der Kontext spielt bei diesem Vorgehen eine wesentliche Rolle. Der Kontext betont und definiert die Sichtweisen auf die Anwendungsfälle und die daraus abzuleitenden Anforderungen. Der Stakeholder-Kontext definiert die Sicht verschiedener Stakeholder und der Systemkontext die für die Systemstruktur wichtige Sicht auf die System-, Subsystem-, Baugruppen- und Komponenten-Ebene (siehe dazu Bild 1). Stakeholder-Anforderungen werden für jede Ebene des Systementwicklungsprozesses (siehe Bild 1, rechts) weiter abgeleitet. Entsprechend können diese auf jeder Ebene identifiziert und dokumentiert werden. Sie beeinflussen dann die Systemstruktur auf der jeweiligen Ebene. Für eine weitergehende Vertiefung des kontextbasierten Requirements-Engineering-Ansatzes nach Holt *et al.* sei auf die Quelle [HPB12] verwiesen.

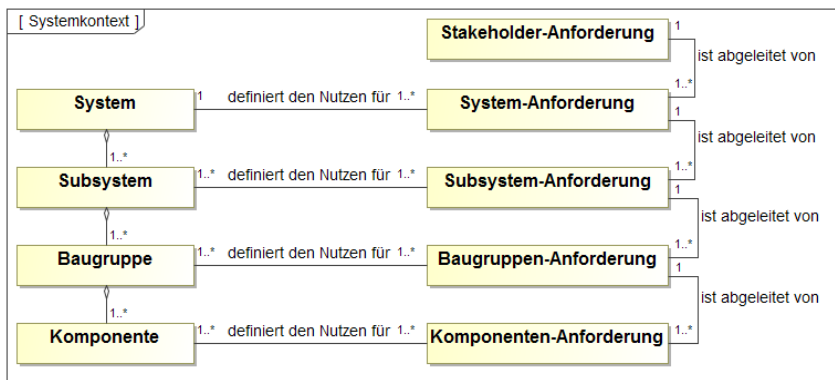


Bild 1. Anforderungen auf den jeweiligen Ebenen des Systementwicklungsprozesses [HPB12]

FAS-Methode: Für die Generierung einer Funktionalen Architektur für Systeme (FAS) haben Lamm und Weilkens [LW15] die FAS-Methode definiert, bei der Aktivitäten als Verfeinerung von Anwendungsfällen spezifiziert werden; d.h. es werden die Aktivitäten der Anwendungsfälle zu funktionalen Gruppen zusammengefasst. Die funktionalen Gruppen und die Abhängigkeiten dieser Gruppen repräsentieren Teile des funktionalen Systems.

Variantenmodellierung: Weilkens [We16] hat für die Modellierung von Varianten eine Methode definiert, welche ein Varianten-Profil in SysML-Notation nutzt und mögliche, so genannte Variationspunkte definiert.

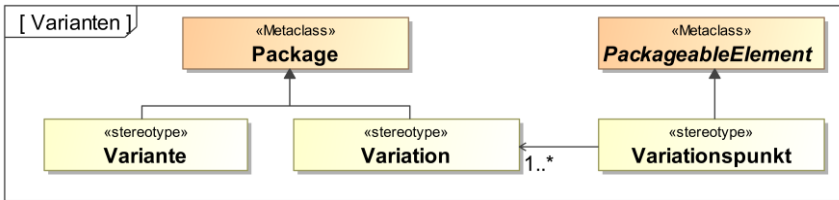


Bild 2. Semantik und Notation einer Variante, einer Variation und eines Variationspunktes [We16]

Bild 2 zeigt dazu die Metaklassen *Package* und *PackageableElement* mit den Stereotyp-Erweiterungen *Variante*, *Variation* und *Variationspunkt*. Ein Stereotyp ist eine Erweiterung vorhandener Modellelemente. Eine Variante ist ein Stereotyp der Metaklasse *Paket* (engl. *Metaclass Package*), welche alle Elemente einer Variationsmöglichkeit enthält. Eine Variation ist ebenfalls ein Stereotyp der Metaklasse *Paket*, welche mehrere Variantenpakete enthält. Ein Variationspunkt ist ein Stereotyp der Metaklasse *PackageableElement* [Ba15] und repräsentiert damit ein Element.

3 Physische Architekturen variantengerecht spezifizieren

Der nachfolgend erläuterte Weg zur Herleitung von variantengerechten physischen Architekturen ergibt sich aus einer Kombination von Methoden für die modellbasierte Systementwicklung unter Verwendung der semi-formalen Modellierungssprache *Systems Modeling Language* (SysML) nach Holt *et al.* [HPB12], nach Weilkiens [We16] und nach Lamm und Weilkiens [LW15] zur Herleitung Funktionaler Architekturen für Systeme (FAS-Methode).

Ein im SiLuFra-Projekt [Si13] erarbeitetes Lösungskonzept, das so genannte Trusted-Forwarder-Konzept, zeichnet sich durch die drei Funktionen Personenidentifikation, Frachtidentifikation und Kopplung von Personen- und Frachtidentitäten zu einem Token aus. In diesem Kapitel wird das Vorgehen zur variantengerechten Spezifikation physischer Architekturen für einen prozessorientierten Projektansatz exemplarisch anhand der Funktion zur Frachtidentifikation erläutert. Es umfasst folgende Aktivitäten:

Projektkontext beschreiben: Alle relevanten Informationen aus dem Kontext des Projektes werden beschrieben, um die Ideen, Ziele und Randbedingungen festzulegen.

Betrachtet man den Projekt-Kontext des SiLuFra-Projektes, kann festgehalten werden, dass das grundlegende Ziel verfolgt wird, die Sicherheit der Luftfracht-Transportkette zu erhöhen und dabei gleichzeitig die Effizienz der wertschöpfenden Prozesse zu steigern oder mindestens zu erhalten. Um dieses Ziel zu erreichen, wurde eine Ist-Aufnahme und -analyse der Prozesse gemäß den folgenden Schritten modellbasiert durchgeführt:

- Die bestehenden Prozesse wurden in Form von Aktivitäten als Aktivitätsdiagramme modelliert.
- Die Funktionen wurden aus der heute bestehenden physischen Architektur abgeleitet und in eine funktionale Architektur überführt.
- Die Funktionen wurden den Aktivitäten zugeordnet und analysiert.

Anschließend wurden an ausgewählten Stellen im Transportprozess Prozessoptimierungen vorgenommen, von welchen in diesem Beitrag eine dieser Optimierungen anhand des so genannten Trusted-Forwarder-Konzepts erläutert wird.

Systemkontext spezifizieren: Das Modellieren des Systemkontextes umfasst die Subaktivitäten: Systemakteure identifizieren, System- und Akteur-Informationsfluss modellieren und Systeminteraktionspunkte identifizieren. Dabei beinhaltet die Systemgrenze alle Systemakteure als direkte Interaktionspartner, für die Schnittstellen entwickelt werden müssen. Damit stellen die Systemakteure eine Untergruppe der ermittelten Stakeholder dar.

Das Trusted-Forwarder-Konzept hat zum Ziel die Transportprozesse im so genannten Vorlauf, d.h. vom Versender bis zum Flughafen, zu verbessern. Dazu gehören die Prozesse der drei Akteure *Known Consignor* (dt. bekannter Versender), *Forwarder* und *Cargo Handling Agent*. Aufgrund des im Projekt vordergründig geforderten Sicherheitsaspektes wurden auch die Prozesse des so genannten Akteurs *Threat* (dt. Bedrohung), d.h. eines möglichen Angreifers, bei der Analyse und Entwicklung des Systems berücksichtigt. Die Ermittlung aller Stakeholder erfolgte dabei gemäß [KG15].

Anforderungen ermitteln: Für den Projekterfolg ist es essentiell, dass die Bedürfnisse aller Stakeholder ausreichend erfüllt werden. Deshalb ist es nötig, sämtliche Personen und Institutionen, die Anforderungen an das System stellen bzw. Interesse am System haben, identifiziert zu haben. Folglich werden bei allen identifizierten Stakeholdern die jeweiligen Anforderungen abgefragt, dokumentiert und strukturiert. Für den hinsichtlich der Sicherheit der Luftfracht-Transportkette im Fokus stehenden Akteur *Threat* wurden zur Abschwächung potentieller Bedrohungen Maßnahmen in Form von sogenannten *Security Requirements* mit Hilfe des modellbasierten Security-Engineering-Prozesses gemäß [HG15] spezifiziert.

Zu den im Trusted-Forwarder-Konzept aus den spezifizierten *Security Requirements* (siehe Bild 3) abgeleiteten funktionalen Anforderungen (engl. *Functional Requirements*) gehören:

- ✓ Identifikation von Personen (engl. *Person Identification*)
- ✓ Identifikation von Fracht (engl. *Cargo Identification*)
- ✓ Kopplung von Personen- und Frachtidentitäten zu einem Token (engl. *eID Combination*)

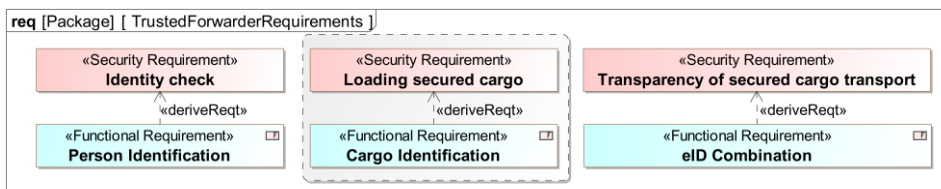


Bild 3. Aus den Sicherheitsanforderungen (engl. *Security Requirements*) für das Trusted-Forwarder-System abgeleitete funktionale Anforderungen (engl. *Functional Requirements*).

Anwendungsfälle modellieren: Anwendungsfälle beschreiben Anforderungen mit den verschiedenen Sichtweisen der Stakeholder. Die o.g. Anforderungen resultieren aus dem allgemeinen Anwendungsfall „Fracht sicher und effizient transportieren“.

Funktionale Blöcke und Aktivitäten definieren: Eine Verfeinerung für jeden Anwendungsfall wird durch die Definition von Aktivitäten vorgenommen. Um jede Aktivität einem entsprechenden funktionalen Block zuordnen zu können, werden funktionale Blöcke definiert. Für den Anwendungsfall „Fracht sicher und effizient transportieren“ wird u.a. die Aktivität „Fracht identifizieren“ und die funktionale Gruppe *Cargo Identification* definiert.

Beziehungen zwischen funktionalen Blöcken und Aktivitäten spezifizieren: In einer Abhängigkeitsmatrix werden die Beziehungen zwischen den funktionalen Blöcken und Aktivitäten hergestellt (siehe Bild 4).

Functional Groups	Cargo Acceptance	Cargo Identification	Cargo Transfer	Cargo Transport	Cargo Unloading	eID Combination	eID Combination Check	Functional System	I/O Cargo	I/O CargoHandlingAgent	I/O Haulier	I/O KnownConsignor	I/O LBA	I/O Regulated Agent	Person Identification	Person Legitimation	Person Legitimation Check
Cargo Acceptance	1																
Cargo Identification	1	1															
Cargo Transfer	1		1														
Cargo Transport	1			1													
Cargo Unloading	1				1												
eID Combination	2					2											
eID Combination Check	2						2										
Functional System								5									
I/O Cargo									5								
I/O CargoHandlingAgent										9							
I/O Haulier											8						
I/O KnownConsignor												6					
I/O LBA													1				
I/O Regulated Agent														10			
Person Identification															2		
Person Legitimation																1	
Person Legitimation Check																	1

Bild 4. Eine Abhängigkeitsmatrix stellt die Beziehungen zwischen den funktionalen Blöcken und den Aktivitäten dar.

Funktionale Architektur des Systems modellieren: Es wird die funktionale Architektur des funktionalen Systems in einem internen Blockdiagramm modelliert. Dafür wird die Abhängigkeitsmatrix als Basis herangezogen, um Ports einzelner *Parts* sowie Konnektoren zu spezifizieren.

Das Bild 5 zeigt, dass der *Part* „:Cargo Identification“ mit drei Ports, die mit Konnektoren mit anderen Ports verbunden sind, Teil der funktionalen Architektur ist. Darüber hinaus hat dieser *Part* keine direkte Verbindung zu den *Parts* :Person Identification und :eID Combination. Diese Information ist für einen Entwickler wesentlich, weil dadurch die Entscheidungsfindung für eine physische Architektur unterstützt werden kann.

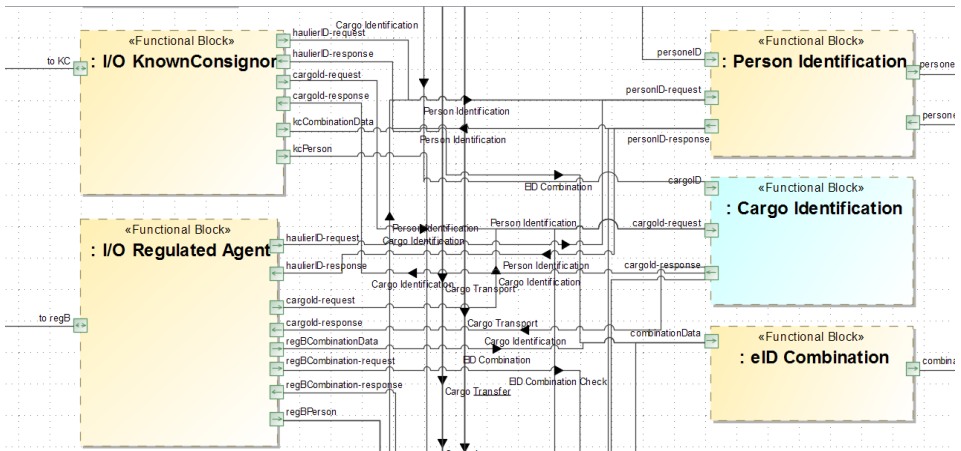


Bild 5. Repräsentativer Ausschnitt der funktionalen Architektur des Trusted-Forwarder-Systems mit den *Part Properties* der funktionalen Blöcke *Person Identification*, *Cargo Identification* und *eID Combination* sowie *I/O Known Consignor* und *I/O Regulated Agent*.

Anforderungen auf jeder Ebene des Systementwicklungsprozesses definieren und dokumentieren: Während des Systementwicklungsprozesses werden von allen Stakeholder-Anforderungen die System-, Subsystem-, Baugruppen- und Komponentenanforderungen abgeleitet (siehe Bild 1). Die entsprechenden Anforderungen für die Frachtidentifikation sind nachfolgend in Bild 6 dargestellt.

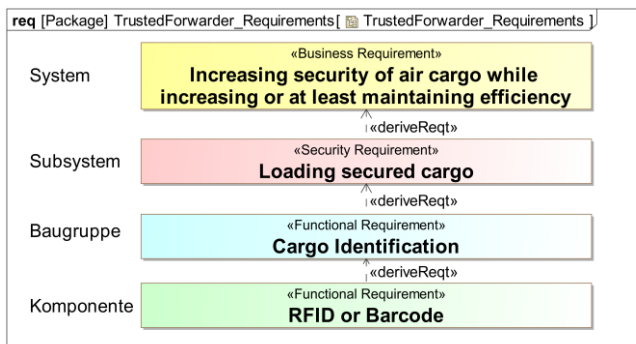


Bild 6. Anforderungen des Trusted-Forwarder-Systems auf den jeweiligen Ebenen des Systementwicklungsprozesses für die Funktion „Cargo Identification“.

Physische Architekturen variantengerecht spezifizieren: In diesem Schritt wird aus der spezifizierten funktionalen Architektur unterstützend eine physische Architektur variantengerecht wie folgt spezifiziert:

- i. Die Variationspunkte einer Variante werden identifiziert. Dabei ist zu berücksichtigen, dass die Anzahl an Variationspunkten möglichst klein bleibt, um ein

- hohes Maß an Standardisierung zu erreichen. Als Resultat entstehen damit gewinnbringende Varianten.
- ii. Das funktionale System wird gemäß der Variantenmodellierung nach Weilkien [We16] (siehe Bild 2) strukturiert. Dabei entsteht eine Paketstruktur bestehend aus Variationen und Varianten mit Variationspunkten. Eine Variation ergibt sich aus dem funktionalen System, Varianten aus den Teilen des funktionalen Systems.
 - iii. Jeder Variationspunkt einer gewinnbringenden Variante wird jeweils durch eine physische Architektur (internes Blockdiagramm) repräsentiert. Im internen Blockdiagramm wird bei der Erstellung der physischen Architektur, welche durch die *Constraints* der Komponenten-Anforderungen getrieben wird, sichtbar, welche *Parts* Variationspunkte sind und welche *Parts* zum Basis-Modell gehören.
 - iv. Im nächsten Schritt wird definiert mit welchen Komponenten sich die Variationspunkte realisieren lassen. Den Komponenten wird im Modell ebenfalls der Stereotyp <<Variation point>> zugewiesen. Dadurch wird in den internen Blockdiagrammen kenntlich gemacht, welche Komponenten variieren können.

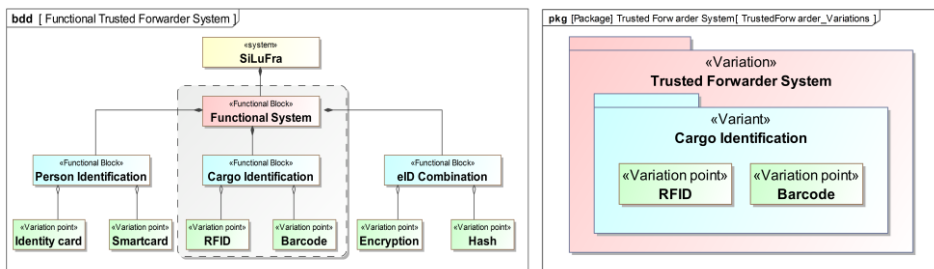


Bild 7. Variationspunkte (links, unten) und die Variation (rechts) „Trusted Forwarder System“ mit der Variante „Cargo Identification“ mit den Variationspunkten „RFID“ und „Barcode“.

Im Bild 7 links ist das funktionale System mit dem System-Teil *Cargo Identification* dargestellt. Dort sind in der untersten Zeile die Variationspunkte *RFID* und *Barcode* abgebildet, da diese beiden Technologien (welche für eine eingängigere Darstellung lediglich Platzhalter für die dahinter stehenden Funktion für Auto-ID-Verfahren sind) bisher zur Frachtidentifikation eingesetzt werden. Die (i.) variantengerechte Strukturierung des funktionalen Systems, die für die (ii.) Ableitung der physischen Architektur herangezogen wird, ist in Bild 7 rechts dargestellt. Das funktionale System selbst ist eine Variation mit dem Namen *Trusted Forwarder System*. Diese Variation beinhaltet die Variante *Cargo Identification* mit dem zwei Variationspunkten *RFID* und *Barcode*.

Unter Berücksichtigung der beiden ermittelten Variationspunkte für die Funktion der Frachtidentifikation entstehen (iii.) zwei physische Architekturen (siehe Bild 8). Die ermittelten Komponenten wie *RFID Reader* und *Barcode Reader*, welche die

Variationspunkte *RFID* bzw. *Barcode* repräsentieren, werden (iv.) dem Stereotyp <<Variation Point>> zugeordnet.

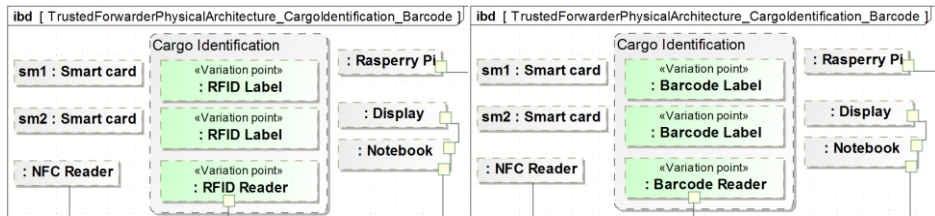


Bild 8: Repräsentative Ausschnitte der physischen Architekturen für Systeme für die Variante *Cargo Identification* mit den Variationspunkten *RFID* (links) und *Barcode* (rechts).

Die Auswahl der Komponenten hängt dabei von den an die Frachtidentifikation gestellten Anforderungen ab (siehe u.a. auch Bild 7), welche es zu erfüllen gilt. Es ist anzumerken, dass die Erarbeitung dieser physischen Architekturen eine Entwurfs-Aufgabe geblieben ist. Es wurden beim Entwurf Entscheidungen über die Örtlichkeit von Komponenten, die Umsetzung von aus den *Security Requirements* resultierenden *Constraints* und über die Anzahl an Hardware-Ressourcen getroffen. Physische Architekturen variantengerecht aus FAS zu spezifizieren ist ein zielgerichtetes Vorgehen, um den Entwickler bei seiner Entscheidungsfindung für eine passgenaue physische Architektur zu unterstützen.

Dieses Vorgehen wurde im Rahmen des SiLuFra-Projektes erarbeitet, verifiziert und anhand des entworfenen Trusted-Forwarder-Systems in einem Feldversuch validiert. Dafür wurden verschiedene physische Architekturen aus FAS variantengerecht spezifiziert, sodass es beispielsweise möglich war innerhalb kürzester Zeit zwei physische Architekturen mit großem Standardisierungsanteil für die Variationspunkte *RFID* und *Barcode* prototypisch in ein System zu überführen, zu simulieren und zu testen. Aufgrund der Standardisierung wurden Schnittstellen definiert, so dass *RFID* und *Barcode* problemlos gegeneinander ausgetauscht werden konnten. Mit Hilfe dieser Ergebnisse konnte eine optimierte physische Architektur für die sichere Luftfracht-Transportkette mit erhöhter Sicherheit und mehr Effizienz abgebildet werden.

4 Zusammenfassung

Prozesse mit bereits bestehenden physischen Architekturen, wie beispielsweise bei der sicheren Luftfracht-Transportkette, lassen sich häufig optimieren. In diesem Beitrag haben wir gezeigt, wie eine Optimierung von Prozessen ausgehend von einer bestehenden physischen Architektur durch Abstraktion zu einer funktionalen Architektur und Reorganisation vorgenommen werden kann, ohne dabei innerhalb physischer Grenzen restriktiv festgelegt zu sein. Ausgehend von der funktionalen Architektur, die mit der FAS-Methode entwickelt und modellbasiert dokumentiert wurde, wurden auf jeder Ebene des Systementwicklungsprozesses Anforderungen weiter abgeleitet und es wurden Variationen und Varianten mit Variationspunkten identifiziert. Im Copyright © 2016 Sylvia Melzer. Zur Veröffentlichung und Nutzung durch GfSE und die mit ihr verbundenen Organisationen freigegeben.

Entwurfsprozess wurden Komponenten zur Umsetzung von Variationspunkten unter Zuhilfenahme weiterer Anforderungen (z.B. Security-Anforderungen, Örtlichkeit von Komponenten und Hardware-Anforderungen) ausgewählt. Anschließend wurde pro Variationspunkt eine physische Architektur erstellt, wobei durch die variantengerechte Strukturierung des Modells kenntlich gemacht wurde, welche Elemente dem Basismodell zuzuordnen sind und welche nicht. Im Ergebnis steuert damit das variantengerechte Vorgehen die Reduktion der Lösungsmenge physischer Architekturen durch den gewinnbringenden Einsatz der funktionalen Architektur für Systeme (FAS), weil durch die Reorganisation von Funktionen mit Hilfe der FAS-Methode ein optimierter Entwurf für die physische Architektur gefunden werden kann. Dieses Vorgehen wurde durch eine prototypische Umsetzung, Verifikation und Validierung des entworfenen Trusted-Forwarder-Systems innerhalb des Projektes SiLuFra [Si13] nachgewiesen.

Danksagung

Das hier vorgestellte Vorgehen entstand im Rahmen des Forschungsprojektes SiLuFra (Sichere Luftfracht-Transportkette: Konzepte, Strategien und Technologien für sichere und effiziente Luftfracht-Transportketten) [Si13]. Dieses Projekt wurde durch das Bundesministerium für Bildung und Forschung (BMBF) innerhalb des Programms „Forschung für die zivile Sicherheit“ zum Themenfeld „Sicherheit im Luftverkehr“ gefördert. Der Feldversuch erfolgte in Zusammenarbeit mit der Luft-Hafen Umschlag GmbH & Co KG (Herrn Dr. Vogt), der Hamburger Logistik Institut GmbH (Herrn Dahlhaus), der Sovereign Speed GmbH (Herrn Naber) und der a.hartrodt (GmbH & Co) KG (Herrn Bruhns). Für die diese Unterstützung danken die Autoren sehr herzlich.

Literaturverzeichnis

- [Ba15] Bahns, T.; Melzer, S.; God, R.; Krause, D.: Ein modellbasiertes Vorgehen zur variantengerechten Entwicklung modularer Produktfamilien, Tagungsband zum Tag des Systems Engineering (Hrsg.: Muggeo, C.; Schulze, S. O.), S. 141-150, 2015.
- [HG15] Hintze, H.; God, R.: Using Model-based Security Engineering in the Development of Complex Aircraft Cabin Systems, SAE 2015 AeroTech Congress & Exhibition, Paper 15ATC-0065, 2015
- [HP08] Holt, J.; Perry, S.: SysML for Systems Engineering. The Institution of Engineering and Technology, London, 2008
- [HPB12] Holt, J.; Perry, S.; Brownsword, M.: Model-Based Requirements Engineering. The Institution of Engineering and Technology, London, 2012.
- [KG15] Kiehl, T.; God, R.: Modeling Top-Level Requirements for a Tangible User Interface in the Aircraft Cabin, AST 2015 - 5th international workshop on aircraft system technologies, S. 275-284, Hamburg, 2015.
- [LW15] Lamm, J.G.; Weikiens, T.: Happy Birthday! 5 Jahre Funktionale Architekturen nach FAS, Tagungsband zum Tag des Systems Engineering (Hrsg.: Chr. Muggeo, S.O. Schulze), S. 59-68, 2015.
- [Me14] Melzer, S.; God, R.; Kiehl, T.; Möller, R.; Wessel, M.: Identifikation von Varianten durch Berechnung der semantischen Differenz von Modellen, Tagungsband zum Tag des Systems Engineering (Hrsg.: Maurer, M.; Schulze, S. O.), S. 279-288, 2014.
- [Si13] Sichere Luftfracht-Transportkette: Konzepte, Strategien und Technologien für sichere und effiziente Luftfracht-Transportketten. <http://www.silufra.de>
- [We16] Weikiens, T.: Variant Modeling with SysML. MBSE4U Booklet Series, 2016.