



Leseprobe

Tag des Systems Engineering

Verteiltes Arbeiten mit ganzheitlicher Kontrolle (Print-on-Demand)

Sonstiger Urheber Christian Tschirner
Herausgegeben von Sven-Olaf Schulze, Christian Muggeo

ISBN (Buch): 978-3-446-44729-5

ISBN (E-Book): 978-3-446-44728-8

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-44729-5>

sowie im Buchhandel.

Vorwort

Sehr geehrte Teilnehmer/Innen,

Ich möchte hier gleich am Anfang die Chance nutzen und der FAS-Arbeitsgruppe (Funktionale Architekturen für Systeme) zum fünfjährigen Bestehen gratulieren und in diesem Zusammenhang auf die vielen Produkte hinweisen, die in diesen Jahren entstanden sind. Neben der jährlichen Teilnahme mit Beiträgen am GfSE Workshop, dem TdSE und Demonstrationen am GfSE Stand hat die Arbeitsgruppe ein Kochbuch, Beiträge in anderen Veröffentlichungen und als neuestes Ergebnis ein FAS Plugin veröffentlicht. Das Plugin ist ein Open-Source Element, das für verschiedene Software Werkzeuge benutzt werden kann.

Neben diesem besonderen Ereignis ist festzustellen, dass es in diesem Jahr wieder eine weitere Fokussierung und Vertiefung auf das Thema MBSE bei den Einreichungen gegeben hat. Die Inhalte werden immer spezifischer und die Beispiele von erfolgreichem Einsatz der Methode nehmen auch in der Industrie zu. MBSE verlässt die Forschung und erobert die Anwendung in unterschiedlichen Industrien. Neben dem Standbein der Luft- und Raumfahrt kommen nun auch Anwendungen aus dem Automobilbau, Schiffbau und dem Anlagen- und Maschinenbau. Viele dieser erfolgreichen Beispiele finden sich in diesem Tagungsband.

Zu den sehr guten Erfahrungsbeiträgen aus der Industrie nimmt die Anzahl der Hochschulen zu, die einen wissenschaftlichen Beitrag einreichen. Die Vielfalt der Hochschulen und Institute, die Systems Engineering mehr in die Lehre und in die Ausbildung aufnehmen ist für mich auch ein Indiz, dass es nun branchenübergreifend immer wichtiger und interessanter wird. Neben den klassischen Ingenieurausbildungen finden sich auch MBSE Ansätze in den Studiengängen der Architektur wieder. Auch ist das Interesse der Hochschulen gewachsen, sich ein Äquivalenzzertifikat zum „Certified Systems Engineer (GfSE)“[®] auf den Masterstudiengang ausstellen zu lassen.

In diesem Sinne wünsche ich Ihnen auch in diesem Jahr erfolgreiche Tage und eine interessante Lektüre zum Systems Engineering. Wir wollen wieder den Austausch unterstützen, und den Wissens- und Erfahrungstransfer zwischen Industrie, Forschung und Lehre fördern. Ich möchte mich bei allen Personen

bedanken, die einen Beitrag eingereicht haben und den ehrenamtlichen Helfern und den Sponsoren, die diese Konferenz und diesen Tagungsband ermöglicht haben.

A handwritten signature in black ink, appearing to read 'S. O. Schulze', written in a cursive style.

Sven-Olaf Schulze,
Vorsitzender der GfSE e.V.

Inhaltsverzeichnis

Der Konferenzband enthält sowohl Industriebeiträge (I) als auch wissenschaftliche Beiträge (W).

T1-1 Werkzeuge

- Gestaltung einer Service-Infrastruktur zur Unterstützung von Systems Engineering Anwendern (I) 3
Christian Zingel, Johannes Fritz
- A Systems Engineering Middleware Concept Built for Incremental Collaboration (I) 13
Günter Staub
- Tool Interoperabilität für durchgängiges modellbasiertes Systems Engineering (W) 15
Christoph Bräuchle, Andrea Leitner, Alfred Wallner

T1-2 Integration I

- IT-Integration mit semantischer Vernetzung für durchgängiges Digital Engineering (I) 27
Joachim Caspar
- Integration von Systemmodellen mit fünf fundamentalen Elementtypen (I) 37
Oskar von Dungern
- Integration von fachbezogenen Ingenieuraktivitäten (W) 47
David Endler

T1-3 FAS(4M)

- Happy Birthday! 5 Jahre Funktionale Architekturen nach FAS (W) 59
Jesko G. Lamm, Tim Weilkiens

Fortschrittsbericht zur modellbasierten Unterstützung der Konstrukteurstätigkeit durch FAS4M (W) <i>Georg Moeser, Christoph Kramer, Martin Grundel, Michael Neubert, Stephan Kümpel, Axel Scheithauer, Sven Kleiner, Albert Albers</i>	69
--	----

T1-4 Integration II

Modellbasierte Konzeption von Benutzerschnittstellen im Entwicklungsprozess von mechatronischen Systemen (W) <i>Christoph Richter, Georg Hackenberg, Peter Stich, Gunther Reinhart</i>	81
--	----

Systems Engineering Handbook 4.0 – Eine neue Ära? (I) <i>Jan von Tongelen</i>	91
--	----

T1-5 Integration III

Skizzen mit formalisierten Informationen anreichern (W) <i>Martin Grundel, Jutta Abulawi, Joscha Loewner</i>	99
---	----

Modelltransformation und Big Data – Die nächsten Herausforderungen im modellbasierten Systems Engineering (I) <i>Oliver Alt</i>	109
---	-----

T2-1 Varianten

Variable Komponenten- und Systemmodelle für effiziente Kommunikation zwischen unterschiedlichen Organisationen (I) <i>Andreas Korff</i>	115
---	-----

Variantenplanung in der Produktarchitektur: Erfahrungen und Best Practices aus der Nutzfahrzeug-Entwicklung (I) <i>Martin Schumacher, Sebastian Märkl, Mark Gilbert, Matthias Kreimeyer</i>	125
---	-----

Ein modellbasiertes Vorgehen zur variantengerechten Entwicklung modularer Produktfamilien (W) <i>Tammo Bahns, Sylvia Melzer, Ralf God, Dieter Krause</i>	141
--	-----

T2-2 Vorgehen

- Modellbasierte Entwicklung und vernetzter Produktentstehungsprozess im Maschinenbau (I) 153
Gerhard Pregitzer, Alexander Blumör, Dr. Sven Kleiner, Michael Neubert, Robert Hämisch
- mecPro² - Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysML (W) 163
Martin Eigner, Thomas Dickopf, Tim Schulte, Marc Schneider
- Vom Systemmodell zu disziplinspezifischen Modellen und zurück (I) 173
Joachim Schmitz, Markus Fockel

T3-1 Einführung von (MB)SE

- Methode zur Wirtschaftlichkeitsbetrachtung der Einführung von Systems Engineering (I) 185
Niklas Hochstein, Marcus Krastel, Martin Langlotz, Oliver Trendel
- Gedanken zur Einführung des System Engineering in mittelständische Unternehmen (I) 195
Thomas Franzen
- Anlagenbauer reagieren zurückhaltend auf aktuelle Trends im Engineering - eine empirische Studie (W) 205
Philipp Gölzer Michael Gepp

T3-2 Kosten

- A Framework to Assess the Cost Impact of Organization and Processes in Complex Systems (W) 217
Michael Roth, Laura Kronfeldner, Daniel Kasperek, Udo Lindemann
- Extended Cost Analysis with Systems Engineering Considerations (W) 227
Christian Schmied, Georg Reinbold, Rachid Amekrane, Eduard Igenbergs, Markus Mörtl, Udo Lindemann

Systems Engineering im Marktwirtschaftlichen Kontext (I) <i>Dieter Scheithauer</i>	237
---	-----

T3-3 Methoden

Using Agile Methods to Improve Efficiency in Requirements Engineering (I) <i>Colin Hood</i>	249
--	-----

Structured analyzing method for design processes of large, complex mechatronic systems applied on a metallurgical plant (W) <i>Andreas Kellner, Sandra Ringhofer, Peter Hehenberger, Stefan Hötzing</i>	257
--	-----

T3-4 PLM

Model Based Systems Engineering und PLM, warum und wie dies bei der Entwicklung von Maschinen zusammengehört (I) <i>Andreas Gallasch, Reimar Schmidt, Jürgen Leiprecht, Thorsten Müller</i>	269
--	-----

Die Rolle von MBSE und PLM im Industrial Internet (W) <i>Michael Pfenning, Christian Muggeo</i>	279
--	-----

T3-5 Modellierung

SystemSketcher – Entstehung eines anwenderorientierten Ansatzes zur interdisziplinären Systemmodellierung (W) <i>Florian Munker, Albert Albers</i>	291
---	-----

Einsatz von Model-based Systems Engineering in der Automobil Industrie (W) <i>Markus Brandstätter, Carolin Eckl</i>	301
--	-----

T4-1 Projekte

- Beschleunigt oder bremst MBSE die Entwicklung von 500 Kilowatt Bremsleistung? (I) 313
Thomas Rogalski
- Pragmatisches Systems Engineering in einem Großprojekt mit Einschränkungen (I) 323
Christian Weibel, Simon Darting, Mario Schmitt, Thomas Kleinberger, Rudolf Braun, Jörg Weber
- Solide Anforderungen dank ReqIF im europäischen Schienenverkehr (I) 333
Michael Jastram, Moritz Dorka

T4-2 Safety

- System Safety in SysML (W) 345
T. Weilkiens, A. Berres, D. Endler, A. Haarer, C. Lalitsch-Schneider, M. Krammer, H. Martin
- Wie gelebtes System Engineering helfen kann, die Anforderungen von Safety Normen umzusetzen (I) 355
Gerd Untermeierhofer, Gerson Laber
- Systementwurf unter Einbeziehung funktionaler Sicherheit bei automobilen Steuergeräten (I) 365
Jan Meyer, Markus Fockel, Jörg Holtmann

T4-3 Modularisierung

- Ein Leitfaden zur marktorientierten top-down Modularisierung im Maschinen- und Anlagenbau (W) 377
Daniel Kasperek, Michael Roth, Christopher Lozano, Udo Lindemann
- Ein Ansatz zur formalen, inhaltlichen Konsistenzprüfung von mechanischen Systemen (W) 387
Martin Grundel, Jutta Abulawi

T4-4 Test / Verifikation I

Qualitätssicherung der Traceability Modelle (W) 399
Atakan Sünnetcioglu, Elisabeth Brandenburg, Uwe Rothenburg, Rainer Stark

Systems Engineering für die Entwicklung der Theorie zu Verlässlichkeit von Systemen (W) 409
Nadine Schlüter, Petra Winzer

T4-5 Test / Verifikation II

Modellzentrisches System- und Testengineering mit maximaler Testautomatisierung im Labor für große mechatronische Systeme (I) 419
Helmut Götz, Dr.-Ing. Tobias Jäger, Dr.-Ing. Hans-Georg Langer, H. Fischer

Mathematische Dimension im Entwurf komplexer Systeme (W) 429
Samuel Vogel, Stephan Rudolph

T1-1 Werkzeuge

Gestaltung einer Service-Infrastruktur zur Unterstützung von
Systems Engineering Anwendern (I)

Christian Zingel, Johannes Fritz

A Systems Engineering Middleware Concept Built for Incremental
Collaboration (I)

Günter Staub

Tool Interoperabilität für durchgängiges modellbasiertes Systems
Engineering (W)

Christoph Bräuchle, Andrea Leitner, Alfred Wallner

Gestaltung einer Service-Infrastruktur zur Unterstützung von Systems Engineering Anwendern

Christian Zingel¹, Johannes Fritz²

¹AVL List GmbH, Instrumentation & Test Systems, Hans-List-Platz 1, 8020 Graz, christian.zingel@avl.com

²Virtual Vehicle Research Center, Information & Process Management, Inffeldgasse 21/A, 8010 Graz, johannes.fritz@v2c2.at

Zusammenfassung: Systems Engineering ist die designierte zentrale Entwicklungsmethodik für komplexe technische Systeme der AVL List GmbH. Dies ist das Ergebnis intensiver Methodenforschung und Vorentwicklungsarbeit. Die verfolgten Ziele sind maximierte Kundenorientierung, nahtlose Einbettung von Systemen in ihre Umgebung und die Handhabung der Komplexität mit maximaler Entwicklungseffizienz. Die industrialisierte Ausrollung von Systems Engineering Paradigmen und modellbasierten Methoden und Werkzeugen birgt zahlreiche neue Herausforderungen und erfordert enorme Anstrengungen bei deren Etablierung in der Organisation. Eine essentielle Maßnahme ist die Einrichtung einer geeigneten Infrastruktur für die Anwender-Unterstützung. Deren Konzept wird in diesem Beitrag vorgestellt. Ein Schwerpunkt liegt hierbei auf der Beschreibung der zugrundeliegenden Service Design-Methodik, mit der die derzeit in Umsetzung befindlichen Dienste spezifiziert und mit den Interessensvertretern im Unternehmen abgestimmt wurden. Der Ausblick beleuchtet vor allem die nächsten Schritte bei der Realisierung von Bibliotheksdiensten zur Ermöglichung der effizienten Wiederverwendung und Vernetzung von Systems Engineering-relevanter Information.

1 Einleitung: Systems Engineering in der AVL

Als weltweit größtes, unabhängiges Unternehmen für Entwicklung, Simulation und Test von Antriebssystemen für PKW, LKW und Großmotoren agiert die AVL List GmbH (AVL) in einer hochvernetzten und rasant voranschreitenden Branche. Essentiell für das Unternehmen und zum Vorteil für seine Kunden ist, dass sich mit den Produkten auch deren Entwicklungsmethoden stetig weiterentwickeln. Nur so können die hohen, selbstgesteckten Qualitätsprinzipien erreicht und gewahrt werden.

Ein zentraler Bestandteil davon ist die Umsetzung der *Integrated Open Development Platform* (IODP), eine Entwicklungsplattform für die nahtlose Integration seiner Produkte und Systeme – gleich ob real oder virtuell in Form von Simulationsmodellen – in deren Umgebung [Pun15]. Damit ist insbesondere die Infrastruktur beim Kunden gemeint, bestehend aus Gebäudetechnik, IKT-Systemen und vorhandenen Testsystemen und -geräten von Drittanbietern, darüber hinaus jedoch auch die Service-Infrastruktur bspw. für Wartung und Firmware-Updates. Der Firmeninhaber Prof. List bezeichnet Systems Engineering in diesem Zusammenhang als den Königsweg für „Mastering Speed and Complexity“ – eine Leitlinie der AVL [VVM14a]. Der Ansatz des Systems

Engineering und des Model-Based Systems Engineering (MBSE) in der industriellen Umsetzung wird im Geschäftsbereich der Instrumentierungs- und Testsysteme als optimale Ergänzung angesehen, diese Herausforderungen zu meistern [VVM14b].

Ende 2013 wurde nach acht Jahren intensiver Methodenforschung und -evaluation (vgl. z.B. [DFD+13]) das Programm *Systems Engineering @ AVL* (SE@AVL) ins Leben gerufen, um Systems Engineering unternehmensweit als zentrale Entwicklungsmethodik für komplexe Systeme zu etablieren und zu verankern. Im Rahmen dieses Programms werden einerseits die Geschäftsbereich-übergreifende Kommunikation und Kollaboration durch die Bildung von Synergien in Prozessen, Methodik und IT-Werkzeugketten gefördert und ausgebaut, andererseits werden aber auch spezifische Elemente wie Arbeitsabläufe, Methoden oder IT-Werkzeugen für die verschiedenen Segmente innerhalb der Geschäftsbereiche erarbeitet und vorangetrieben [WFD+14].

Die Anwendung der Systems Engineering-Methodik im industriellen Umfeld impliziert sowohl für die Entwicklungsorganisation als auch das Management einen Paradigmenwechsel und somit auch einen fundamentalen Eingriff in gelebte Denk- und Arbeitsabläufe. Darüber hinaus kommen - teilweise überlagert - neue Prozesse, Methoden und Werkzeuge sowie veränderte Organisationsstrukturen zum Einsatz, die eine enge Begleitung der involvierten Personen in ihrem Lernprozess erfordern. Dies wird zwar in der AVL durch umfassende Schulungs- und Betreuungsmaßnahmen sicher gestellt, jedoch wächst durch die steigenden Anwenderzahlen gleichzeitig der Bedarf nach Zugang zu schneller und effizienter Anwenderunterstützung, damit die Produktivität durch den Lernprozess geringstmöglich beeinträchtigt wird (vgl. Bild 1). Hierzu entsteht eine umfassende Service-Infrastruktur unter Anwendung von *Service Design-* bzw. *Service Engineering-*Methoden.

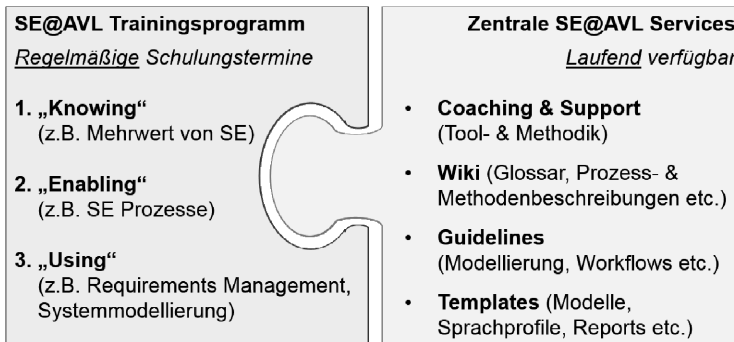


Bild 1: Anwenderunterstützung im Rahmen von SE @ AVL [eigene Darstellung]

Die Service Design- bzw. Service Engineering-Methodik sowie das resultierende Konzept mit ersten realisierten zentralen Diensten werden in den nachfolgenden Kapiteln im Detail vorgestellt. Zuletzt folgt ein Ausblick auf weitere Maßnahmen in Form von Bibliotheksdiensten für die konsistente Wiederverwendung und Vernetzung von Systems Engineering-relevanter Information.

2 Entwicklungsmethodik - Service Design

Umfang und Bedeutung von Service Design (englischsprachiger Raum) bzw. Service Engineering (deutschsprachiger Raum) sind je nach Kontext unterschiedlich belegt, eine eindeutige und allgemeingültige Definition für diesen interdisziplinären Ansatz gibt es nicht, vgl. u.a. [Nis12], [SS11]. Bild 2 zeigt ein Überblicksmodell zu Service Design mit den drei Hauptbestandteilen *Organisation*, *Service Design* und *Kunde* sowie deren Wechselwirkungen [Mor05].

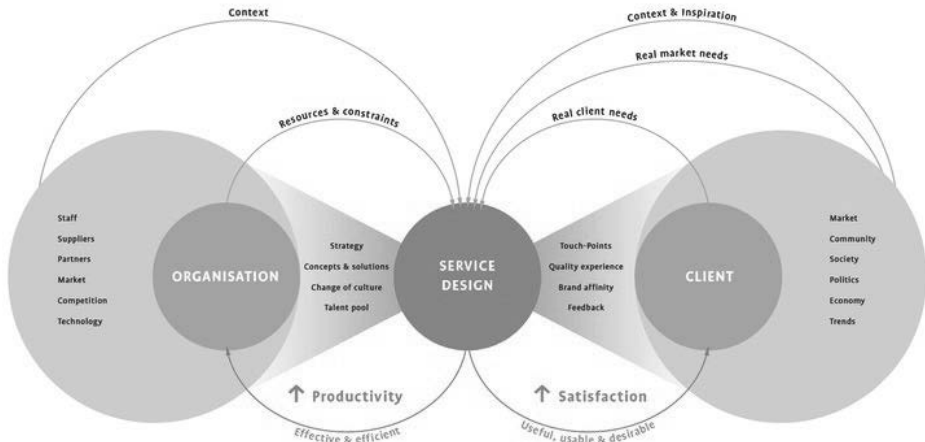


Bild 2: Service Design Overview Modell [Mor05]

Im Kontext dieses Beitrags wird unter Service Design bzw. Service Engineering die systematische Planung, Entwicklung und Gestaltung von Dienstleistungen unter Einbezug der involvierten Personen und der Informations- und Kommunikationstechnologie-Infrastruktur verstanden. Das Ziel ist die nachhaltige Entwicklung von zentral administrierten Anwender-Dienstleistungen, die in vier Problemlösungsschritten vollzogen werden soll, vgl. [Mor05], [Hun15]:

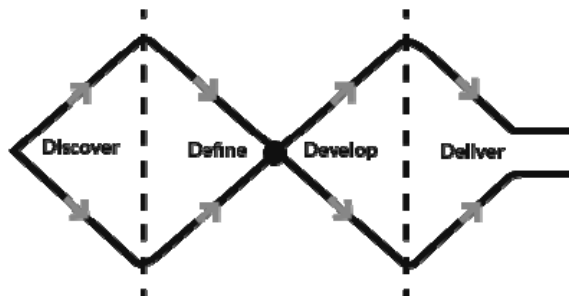


Bild 3: Double Diamond Problem Solving Process Modell [Eng15]

Zur Entwicklung bestmöglicher Services für die Systems Engineering-Anwender in der AVL kamen verschiedene Service Design-Methoden und –Werkzeuge zum Einsatz. Die Vorgehensweise bestand im Wesentlichen aus den folgenden Schritten, vgl. [FD13], [Pla10]:

1. Anforderungserhebung per *Interviewserie* „Fachdisziplinübergreifendes Informationsmanagement & Kommunikation“
2. Extraktion von servicerelevanten Bedürfnissen bei der Anwendung von Systems Engineering-Methoden (*Kundenanforderungen*)
3. Konsolidierung mittels *Customer Journeys / Szenariotechnik*
4. Formalisierung charakteristischer Service-Abläufe mittels *Service Blueprints*
5. Extraktion von *technischen Anforderungen* an die Umsetzung des Service-Konzepts
6. *Technische Konzeption & Implementierung* eines Service-Portals als zentraler Kommunikationskanal mit einem verknüpftem Workflow-Managementsystem zur Abwicklung von Anfragen
7. *Ausrollung* der Services in die Produktivumgebung
8. *Laufende Evaluierung* und *Adaption* der Services

Der erste Schritt der Analysephase bestand in der Durchführung einer qualitativen Interviewreihe mit verschiedenen Interessensvertretern im Unternehmen. Die verfolgte Zielsetzung lag in der Erhebung von Anwenderbedürfnissen hinsichtlich mehrwertbildender Systems Engineering-Methoden bei der fachdisziplinübergreifenden Arbeit in der AVL. Den inhaltlichen Schwerpunkt hierbei bildeten Fragen zu Methoden und Werkzeugen zur Ablage, Bereitstellung und Kommunikation disziplinübergreifend relevanter Information. Da dies ein durchaus breites Spektrum mit teilweise recht unterschiedlich gelagerten Interessen der Befragten in der AVL darstellte, wurden die Interviews als qualitatives Interview, also im Form eines offenen Gesprächs mit unterstützendem Leitfaden durchgeführt, um individuelle Schwerpunkte besser beleuchten zu können, vgl. [DP13].

Im Anschluss an die Interviewreihe wurden die Interviewprotokolle transkribiert, anonymisiert und systematisch ausgewertet. Hierzu wurden Klassifikationen im Vorfeld erarbeitet, die eine Zuordnung einzelner Aussagen zum Gesamtbild ermöglichen sollte. Im Rahmen der Auswertung wurden diese Klassifikationen präzisiert. Als Ergebnis dieses Schritts lagen klassifizierte Kundenanforderungen vor (Schritt 2). Als besonders wichtig erachtet wurden definierte Ordnungsschemata und Regularien bei der Informationsablage sowie umfassende Such- und Filterfunktionalitäten bei der Informationsakquise. Auch die sowohl inhaltliche als auch technische Quervernetzung und Nachverfolgbarkeit (Traceability) von Informationen über verschiedene Aspekte hinweg wurde als sehr bedeutsam eingestuft.

Mit Hilfe der Customer Journey-Methodik wurden basierend auf den Kundenanforderungen die Erwartungshaltung und die Wahrnehmung archetypischer Service-Nutzer

herausgearbeitet. Hier konnten Diskrepanzen zwischen einzelnen Interviewaussagen und dem intendierten Kundenerlebnis des jeweiligen Services visualisiert werden (Schritt 3).

Zur weiteren Detaillierung und Schematisierung wurden für jeden Dienst sogenannte Service Blueprints erstellt, vgl. Bild 4. Sie ermöglichen eine Trennung der Zuständigkeiten in Bahnen (Kunde, Front Office, Back Office, etc.) und eine Workflow-orientierte, technische Repräsentation inklusive der jeweiligen Interaktionskanäle (Schritt 4).

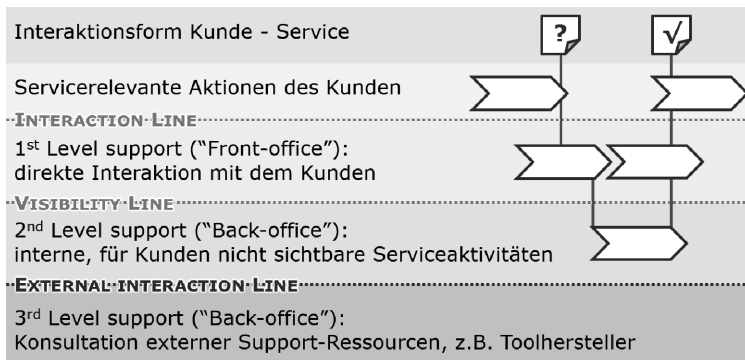


Bild 4: Schema eines Service Blueprints [eigene Darstellung]

Ausgehend von den Service Blueprints wurden technische Anforderungen an die Dienste, deren notwendige Infrastruktur und deren Umsetzung abgeleitet wie bspw. an den Grad an IT-Unterstützung, Automatisierungen oder Interaktionskanäle (Schritt 5).

Damit wurde die Grundlage für die Konzeption und die technische Implementierung des Service-Portals gelegt, das im nächsten Kapitel vorgestellt wird (Schritt 6).

Auf Basis der Implementierung wurde ein Deployment-Prozess gestartet (Schritt 7), der in Folge in einen kontinuierlichen Verbesserungsprozess gemäß dem Deming-Kreis (Plan – Do – Check – Act) mit laufend iterativer Evaluierung von Anwenderakzeptanz und Anwenderbedarfen und entsprechenden Anpassungen bzw. Weiterentwicklungen der Dienste mündet, vgl. [Dem82] (Schritt 8).

3 Die SE@AVL Service-Infrastruktur

Basierend auf den Ergebnissen der Analysephase und den daraufhin konzipierten Diensten zur bestmöglichen Unterstützung der SE-Anwender in der AVL wurde ein SE@AVL Serviceportal als Herzstück der Service-Infrastruktur eingerichtet (siehe Bild 5). Die zugrundeliegende Technologie ermöglicht die Realisierung zahlreicher Online-Dienste für das geführte Selbststudium der Anwender wie ein Wiki-System mit Guidelines, Arbeitsempfehlungen und Beispielen sowie Glossare, Literaturlisten, Querverweise oder FAQ. Darüber hinaus konnte damit auch eine direkte technische Integration mit einem Workflow-Managementsystem zur Abwicklung von Anfragen realisiert werden, das in

der AVL bereits zur Abwicklung von Engineering IT-Anfragen im Einsatz war. Neben dieser Plattform werden als Interaktionskanäle auch Email oder Telefonsupport mit Screensharing angeboten, die ebenfalls mit dem Workflow-Managementsystem zur Abwicklung der Tickets arbeiten.

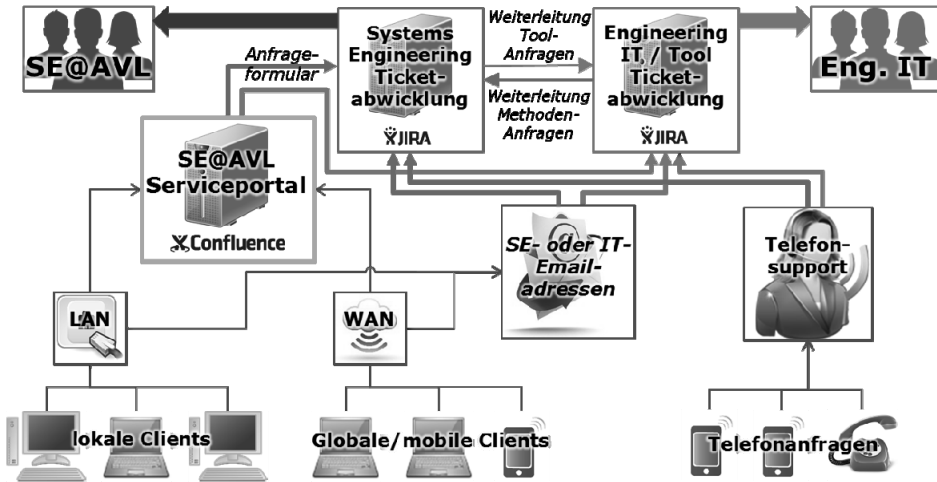


Bild 5: Architektur der zentralen Dienste für die Anwender-Unterstützung [eigene Darstellung]

Eine zentrale Rolle bei der Service-Plattform spielt die Unterstützung von Anwendern modellbasierter Systementwicklung (MBSE). Hierzu werden im Rahmen des Programms SE@AVL zentral entwickelte Modellierungsmethoden in dieser Plattform in Form modular aufgebauter Modellierungs-Guidelines dokumentiert. Hier werden sowohl allgemeingültige, normative Informationen zur Notation und Semantik der eingesetzten Modellierungssprachen (z.B. UML und SysML) als auch bereichs- oder disziplinspezifische Erweiterungen von Sprachen und Methodik beschrieben. Die einzelnen Themen sind in modularen Kapiteln mit Schlagworten versehen strukturiert, um den Anwendern wie gewünscht bestmögliche Such- und Navigationsfunktionen zur Verfügung stellen zu können. An jeder Stelle ist es den Anwendern möglich, Anfragen bei Unklarheiten oder Problemen zu stellen oder Feedback abzugeben, um eine effiziente, kontinuierliche Verbesserung und Weiterentwicklung gemäß dem Deming-Kreis zu ermöglichen.

Die Vermittlung der Systems Engineering-Kompetenzen (vgl. Bild 1) wird durch ein dreistufiges Ausbildungsrahmenwerk sichergestellt, worin das insgesamt sehr breite Inhaltsspektrum zielgruppengerecht strukturiert ist. Die drei Stufen dienen hierbei insbesondere der Berücksichtigung der unterschiedlichen Sichtweisen auf Systems Engineering, beginnend mit einem sehr allgemeinen, aber breiten Spektrum über die Paradigmen des Systems Engineering bis hin zu sehr scharf abgegrenztem, aber detailtiefem Anwendungswissen zu konkreten Methoden und Werkzeugen (vgl. Bild6).

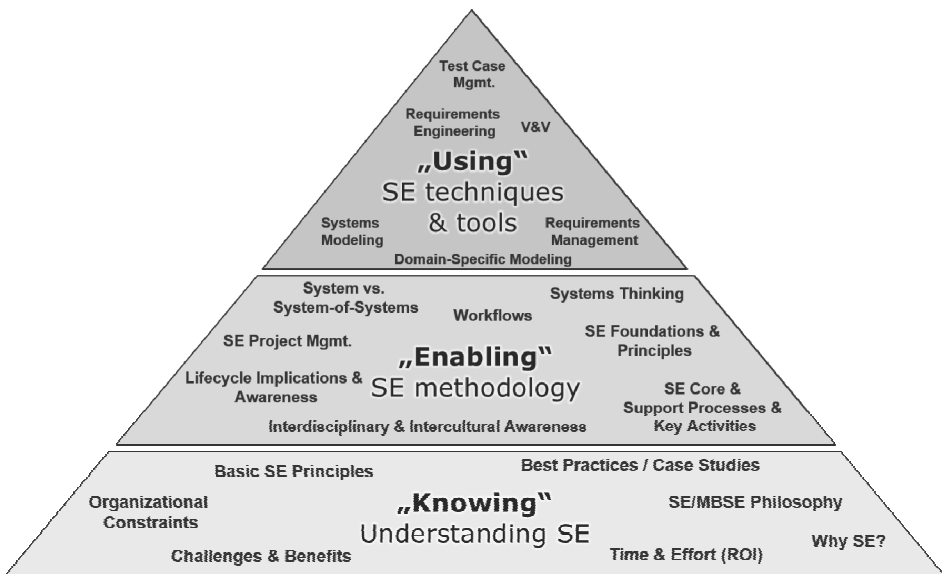


Bild 6: SE@AVL Trainingspyramide

Erst das enge *Zusammenspiel* von *zentralen Diensten* und *dezentralem Coaching* nach einer *fundierten Grundausbildung* ist hinreichend effizient, das für viele Anwender neue Themenfeld Systems Engineering zu operationalisieren – also zu „leben“.

4 Ausblick: Bibliotheksdienste

Aufbauend auf der vorgestellten Service-Infrastruktur ist es möglich, Dienste zur Wiederverwendung von Systems Engineering relevanter Information zu implementieren. Insbesondere die Wiederverwendung von Modellfragmenten und die Bereitstellung von Automatismen bei der Anwendung von MBSE stellen ein hohes Effizienzsteigerungspotential dar. Hierzu gehören die Einrichtung und Wartung von bspw. folgenden Diensten:

1. Kontextspezifische *Profilerverweiterungen* für Modellierungssprachen ermöglichen die Anpassung an die jeweiligen Gegebenheiten durch die Definition spezialisierter Modellierungselemente (wie Entitäten, Relationen oder Attribute in der entsprechenden Domänenterminologie).
2. Vordefinierte *Modell-Templates* für verschiedene Geschäftsbereiche (wie das Produktgeschäft oder den Anlagenbau), die eine auf die jeweils involvierten Entwicklungsaufgaben und Rollen hin angepasste Struktur vorgeben. Diese besteht neben einer Orderstruktur und entsprechenden Profilerweiterungen der Modellierungssprache auch aus vorbereiteten Standard-Diagrammen (Sichten).
3. *Templates* für Prozessdokumente, die teil- oder vollautomatisiert Informationen aus den Modellen extrahieren und unter Anwendung der vordefinierten Modell-

strukturen und –Sichten in eine allgemein zugängliche Form überführen (z.B. Office-Dokumente oder pdf-Dateien).

4. *Granulare Modellierungs-Artefakte* wie Akteure, Schnittstellendefinitionen, SI-Einheiten, Dimensionen, etc. können in einheitlicher Form vorgehalten werden.
5. *Model Assets* stellen höherintegrierte Modellfragmente für die Wiederverwendung dar. Es handelt sich hierbei um z.B. Standardkomponenten oder Baugruppen entweder eigenentwickelt oder auch von Drittanbietern, die in mehreren Entwicklungsprojekten angewendet werden können. Hierzu zählen Mechanikbauteile (z.B. Schrauben, Federn, Dämpfer) oder Elektronikbauteile (Stecker, PCBs, Sensoren, Aktoren), aber auch Software Services niedriger Abstraktionsebenen (als wiederverwendbare Teile von Service-Orientierten Architekturen) oder Software-Interfaces (z.B. Bus-Technologien).

Die Bibliotheksdienste tragen zur Operationalisierung der Ergebnisse AVL-interner Initiativen zur Standardisierung und Modularisierung des Instrumentierungs- und Testsystemportfolios der AVL bei. Das Ziel ist die signifikante Steigerung des Standardisierungsgrades von Bauteilen und Baugruppen (z.B. Schnittstellentechnologien, Sensoren und Aktoren) und damit eine Flexibilisierung der Produktintegration in die somit maximiert variablen Testsysteme (vgl. Bild 7).

Modul-typ	Identisch	Diskret	Parametrisch	Kundenspezifisches Modul
Modul-Design	Eine <u>einzig</u> e Bauform	<u>Zwei oder mehr</u> definierte Bauformen	<u>Basismodell</u> mit definiertem Set anpassbarer Parameter	<u>Kundenspezifisches Design</u>
Varianten	Keine Modulvarianten vorhanden	Modulvarianten entsprechen den Bauformen	Modulvarianten durch Variation der Parameter	Modulvarianten durch Anpassung an Kundenanforderungen
Verwendung	Die Module werden unverändert in verschiedenen Anlagen/Projekten verwendet.	Die Module können im Rahmen der verfügbaren Bauformen den Anforderungen an die Anlage angepasst werden.	Die Module können durch Veränderung der Parameter den Anforderungen an die Anlage angepasst werden.	Die Module können ohne Einschränkungen den Kundenanforderungen angepasst werden.
Standardisierungsgrad				Flexibilität

Bild 7: Modultypen über Grad der Standardisierung bzw. Flexibilität [Ge14]

Kritisch zu betrachten ist bei allen Bestrebungen der Aspekt der erforderliche Pflegeaufwand der vorgestellten Bibliotheksdienste, welcher mit zunehmend hohem Standardisierungsgrad und entsprechender Regulierung sowie zunehmender Verbreitung der Anwendung progressiv steigt. Hier ist das ausgewogene Verhältnis aus Standardisierung, Modularisierung, Anwendungsspektrum („Breite“) und Detaillierung („Tiefe“) entschei-

dender Erfolgsfaktor. Hierbei sind insbesondere die spezifischen Rahmenbedingungen der Instrumentierungs- und Testsysteme der AVL zu betrachten. Wo die Stückzahlen bei einzelnen Mess- und Konditioniergeräten noch einer Kleinserie entsprechen, so sind Testsysteme als Anlagen zu verstehen, die einem sehr hohen Individualisierungsgrad unterliegen. Dennoch ist ein ausgewogenes Verhältnis aus Standardisierung und Flexibilisierung auf Subsystem- oder Bauteilebene auch im hochindividualisierten Anlagenbau ein Instrument mit erheblichen Effizienzsteigerungs- und Kostensenkungspotentialen.

Literaturverzeichnis

- [BS06] Bullinger, H.-J.; Scheer, A.-W.: Service Engineering – Entwicklung und Gestaltung innovativer Dienstleistungen, 2. Auflage. Springer, Berlin, Heidelberg, 2006.
- [Dem82] Deming, W.E.: Out of the Crisis. Massachusetts Institute of Technology, Cambridge 1982.
- [DFD+13] Denger, A.; Fritz, J.; Denger, D.; Zingel, C.; Ölmez, M.; Kissel, M.: Modell-basierter Ansatz zur funktionsorientierten Lenkung von Produkten. In: Produktdatenjournal Nr. 1/2013. ProSTEP iViP e.V., Darmstadt, 2013.
- [DP13] Dresing, T.; Pehl, T.: Praxisbuch Interview, Transkription & Analyse. Anleitungen und Regelsysteme für qualitativ Forschende. 5. Auflage. Marburg, 2013. Quelle: www.audiotranskription.de/praxisbuch (Datum des Downloads: 23.03.2015)
- [Eng15] Webseite der Engine Group: Approach - Engine Service Design. Quelle: <http://enginegroup.co.uk/approach/> (Datum des Downloads: 30.06.2015)
- [FD13] Fritz, J., Denger, A.: Best-Practice-Ansatz zur Erfassung und Modellierung von Stakeholder-Sichten. In: Tag des Systems Engineering – Der Weg zu den technischen Systemen von morgen. Carl-Hanser-Verlag, München, 2013.
- [FDW+15] Fritz, J.; Denger, A.; Weber, J.; Zingel, C.; Denger, D.: Model-Based Systems Engineering für Testsysteme. Hanser Automotive Nr. 5-6 / 2015. Carl Hanser Verlag, München, 2015.
- [Ge14] Gepp, M.: Standardisierungsprogramme als Ansatz zur Steigerung der Wirtschaftlichkeit im industriellen Anlagen-Engineering. Dissertation an der Rechts- und Wirtschaftswissenschaftlichen Fakultät des Fachbereichs Wirtschaftswissenschaften. Friedrich-Alexander-Universität Erlangen-Nürnberg. 2014.
- [Hun15] Hunter, M.: The Design Process: What is the Double Diamond? Design Council, 2015. Quelle: <http://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond> (Datum des Downloads: 07.05.2015)
- [Mor05] Moritz, S.: Service Design – Practical Access to an Evolving Field. , London, 2005. Quelle: <http://www.stefan-moritz.com/#book> (Datum des Downloads: 01.04.2015)
- [Nis12] Nisula J.-V.: Searching for Definitions for Service Design – What do we mean with Service Design? In: Tagungsband 3rd Service Design and Service Innovation Conference. Espoo, Finnland, 2012.
- [Pla10] Plattner, H.: Bootcamp Bootleg. Institute of Design at Stanford, 2010. Quelle: <http://dschool.stanford.edu/wp-content/uploads/2011/03/BootcampBootleg2010v2SLIM.pdf> (Datum des Downloads: 30.03.2015)
- [Pun15] Puntigam, W.: Integrated Functional Development Process – Simulation- or Test-based? In: Tagungsband des 8. Grazer Symposium Virtuelles Fahrzeug (GSVF). Graz, 2015.
- [SS11] Stickdorn, M; Schneider, J.: This is Service Design Thinking: Basics - Tools – Cases. BIS Publishers, Amsterdam, 2011.

- [VVM14a] Kompetenzzentrum Das Virtuelle Fahrzeug Forschungsgesellschaft mbH (Hrsg.): Entwicklung, Simulation und Test von Antriebssystemen (Interview Prof. Helmut List, CEO AVL List GmbH). In: Virtual Vehicle Magazine Nr 18., III-2014, S. 20-22. Graz, 2014.
- [VVM14b] Kompetenzzentrum Das Virtuelle Fahrzeug Forschungsgesellschaft mbH (Hrsg.): Systems Engineering in der Prüfstandsentwicklung (Interview Kai Uwe Voigt, EVP AVL List GmbH). In: Virtual Vehicle Magazine Nr 20|2014, S. 24-25. Graz, 2014.
- [WFD+14] Weber, J.; Fritz, J., Denger, A.; Maletz, M.; Zingel, C.; Denger, D.:: Synergien in der modellbasierten Antriebsstrang- und Testsystem-Entwicklung. In: Tagungsband des 17. SIMVEC - Simulation und Erprobung in der Fahrzeugentwicklung. Baden-Baden, 2014.

A Systems Engineering Middleware Concept Built for Incremental Collaboration

Dr. Günter Staub

PDTec AG, Albert-Nestler-Str. 10, 76131 Karlsruhe

Anmerkung der Redaktion:

Die PDTec AG ist freundlicherweise sehr kurzfristig für einen anderen Beitrag eingesprungen, weshalb kein Paper vorliegt.

Tool Interoperabilität für durchgängiges modellbasiertes Systems Engineering

Christoph Bräuchle¹, Andrea Leitner^{2,3}, Alfred Wallner³,

¹Parametric Technology GmbH, Edisonstr. 8, D-85716 Unterschleißheim,
cbraeuchle@ptc.com

²Kompetenzzentrum – Das virtuelle Fahrzeug Forschungsgesellschaft mbH, Inffeldgasse
21a A-8010 Graz, Andrea.Leitner@v2c2.at

³AVL LIST GMBH, Instrumentation and Test Systems, Hans-List-Platz 1, A-8020 Graz,
Alfred.Wallner@avl.com

Zusammenfassung: Offene Formate und Spezifikationen wie OSLC (Open Services for Lifecycle Collaboration), ReqIF (Requirements Interchange Format) oder FMI (Functional Mockup Interface) wecken Erwartungen nach einer durchgängigen Systems Engineering Werkzeugkette, in der best-of-breed Technologien und Tools unterschiedlicher Hersteller kombiniert werden können.

Im CRYSTAL Projekt wird im öffentlichen Use Case der Automotive Domäne eine solche Werkzeugkette für durchgängiges modellbasiertes Systems Engineering aufgebaut. Hierbei werden unter anderem die Verwaltung von Anforderungen, Tests und Simulationen, sowie die Unterstützung von Co-Simulation mit einer Integrationsplattform abgedeckt.

1 Einleitung

1.1 Motivation

Die Entwicklung komplexer Systeme nimmt einen immer höheren Stellenwert in der Automobilindustrie ein, wobei bis zu 90% der Innovationen durch Software ermöglicht werden. Ein wesentlicher Erfolgsfaktor liegt in der Fähigkeit, diese Systeme bereits im Entwurf so zu konzipieren, dass das Zusammenspiel der enthaltenen Software- und Hardwarekomponenten optimal ausgelegt ist. Herkömmliches, dokumentbasiertes Arbeiten wird dem immer höheren Vernetzungsgrad zwischen den Teilsystemen und Komponenten nicht mehr gerecht – hier sind modellbasierte Techniken und Werkzeuge gefordert, die klassisches Anforderungsmanagement ergänzen und vertiefen. Späte Validierung der Systemfunktionalität resultiert in einem heute kaum mehr vertretbaren Projektrisiko – hier werden Möglichkeiten zur frühzeitigen Simulation von Verhaltensmodellen benötigt. Entsprechende Werkzeugunterstützung ist dabei unumgänglich. Dabei reift die Erkenntnis, dass universelle „All-in-one“ Lösungen mit monolithischer Datenhaltung und proprietären Schnittstellen nicht flexibel genug auf spezifische und sich rasch ändernde Anforderungen reagieren können. Stattdessen sind offene Architekturen gefragt, die den Aufbau von heterogenen Werkzeugketten

ermöglichen und die lückenlose Integration unterschiedlicher disziplin-spezifischer Tools ermöglichen.

1.2 Ansatz

Anhand eines konkreten Beispiels werden die Möglichkeiten und Grenzen offener Spezifikationen zur Tool-Interoperabilität aufgezeigt. Besonderes Augenmerk liegt hierbei auf offenen Schnittstellen, die klassische Lifecycle Management Funktionen – beispielsweise Anforderungsverwaltung – mit modellbasierter Entwicklung verbinden, und die Einbindung unterschiedlicher Tools in die Toolkette ermöglichen.

Die in diesem Artikel vorgestellten Arbeiten und Ergebnisse wurden von den Autoren im Artemis JU Projekt CRYSTAL¹ erarbeitet (Critical Systems Engineering Acceleration). Ein wichtiges Ziel dieses Projekts ist das Etablieren einer offenen Interoperabilitätsspezifikation, welche die Entwicklung sicherheitskritischer Systeme unterstützt. Diese Interoperabilitätsspezifikation setzt sich aus bestehenden Standards zusammen, die im Projekt bei Bedarf erweitert werden und deren Verwendung in „Best-practice guidelines“ für häufige Anwendungsszenarien beschrieben wird. Ein wesentlicher Bestandteil ist dabei die OSLC² Spezifikation (Open Services for Lifecycle Collaboration) zur Unterstützung von Lifecycle Interoperabilität und die FMI Spezifikation (Functional Mockup Interface) zur Kopplung dynamischer Systeme. Aktuell wird auch die Aufnahme von ReqIF³ (OMG Requirements Interchange Format) als Austauschformat für Anforderungen vorangetrieben. Der hier beschriebene Demonstrator wendet diese Spezifikationen und Standards in einem typischen automotiven Szenario an und ermöglicht damit die Betrachtung von Vor- und Nachteilen sowie möglicher Limitierungen. Dieser Artikel beleuchtet schwerpunktmäßig folgende Themen:

- Aufbau einer heterogenen Systems Engineering Toolkette unter Verwendung offener, webbasierter Schnittstellen
- Verbindung etablierter Lifecycle Management Funktionen mit innovativen, modellbasierten Engineering Methoden
- Evaluierung der Möglichkeiten von offenen Interoperabilitätsstandards und –spezifikationen
- Aufzeigen von Verbesserungspotenzialen

2 CRYSTAL Automotive Demonstrator und öffentlicher Use Case

Da im Projekt CRYSTAL die Möglichkeiten offener Tool-Architekturen in teils geheimen Entwicklungsvorhaben der Projektteilnehmer evaluiert werden, hat die Automotive Domäne einen so genannten öffentlichen Use Case definiert, in dem die

¹ www.crystal-artemis.eu/

² <http://open-services.net/>

³ <http://www.omg.org/spec/ReqIF/>

wesentlichen Erkenntnisse aus den nicht-öffentlichen Anwendungsfällen der Industriepartner zusammengefasst sind. Eine schematische Darstellung dieses öffentlichen Use Case, so wie er am Ende des Projektes (April 2016) demonstriert werden soll, findet sich in Abbildung 1.

Neben der Möglichkeit, Szenarien aus dem CRYSTAL Projekt einem breiten Publikum zugänglich zu machen, ist ein wichtiges Resultat dieses öffentlichen Use Cases die Kombination von Teilszenarien der einzelnen industriellen Use Cases, womit auch die Wiederverwendbarkeit der Schnittstellen demonstriert wird. Darüber hinaus werden Tools verschiedener Hersteller miteinander gekoppelt um die Anwendbarkeit der Schnittstellen in einer heterogenen, Multi-Vendoren Toollandschaft festzustellen. In einem realen Anwendungs-Szenario würde ja üblicherweise nur ein einzelnes SysML Modellierungswerkzeug zum Einsatz kommen. Dennoch wurde hier parallel zum SysML Werkzeug PTC Integrity Modeler der Sparx Enterprise Architect für die modellbasierte Erstellung eines Funktionalen Sicherheitskonzepts eingesetzt um so die Herstellerunabhängigkeit der OSLC-Schnittstellen zu demonstrieren.

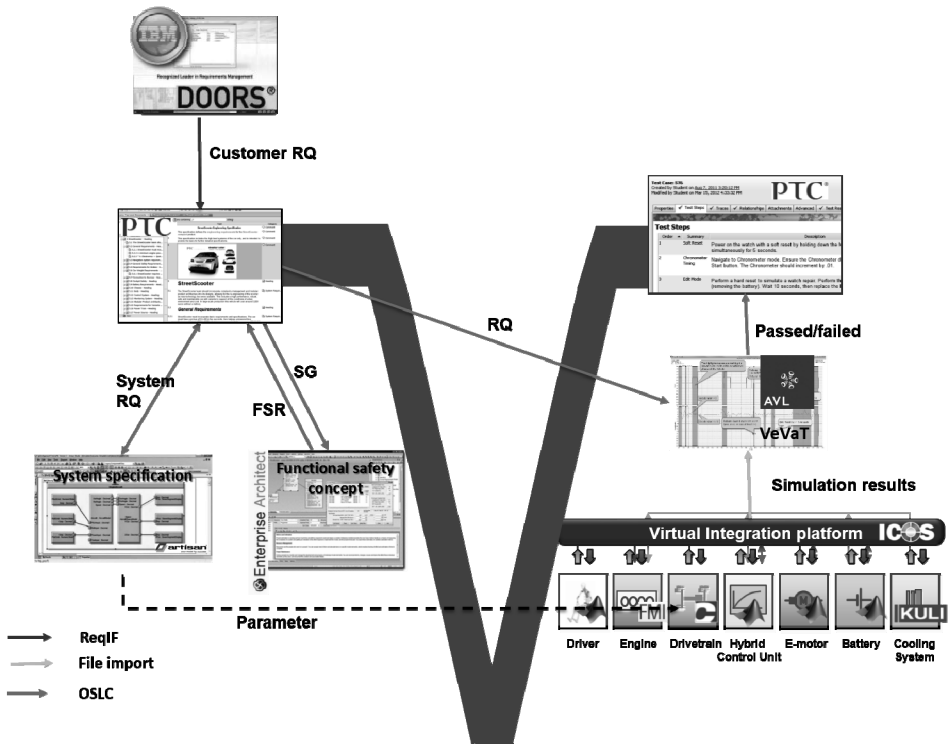


Abbildung 1: Schematische Darstellung des Automotive Demonstrators

2.1 Verwendete Tools und Schnittstellen

Im CRYSTAL Automotive Demonstrator kommen Software Tools unterschiedlicher Hersteller zum Einsatz. Diese Tools bieten teils erheblich mehr Funktionalität und Schnittstellen, als die hier verwendeten. In Tabelle 1 sind lediglich diejenigen aufgeführt, die auch konkret im öffentlichen Use Case verwendet werden.

Tool	Einsatzzweck im Demonstrator	Verwendete Schnittstellen
<i>PTC Integrity Lifecycle Manager</i>	Verwaltung der Anforderungen und Testfälle im Entwicklungsprozess	<ul style="list-style-type: none"> • ReqIF (Requirements Interchange Format) zum Import des Kundenlastenhefts • OSLC-RM (Requirements Management) Provider zur Anbindung von MBSE Tools • OSLC-Asset Client zur Kopplung von Systemlastenheft und Sicherheitskonzept • OSLC-QM (Quality Management) Provider zur Anbindung von Testausführungs-Tools
<i>IBM DOORS</i>	Quelle des Kundenlastenhefts	ReqIF zur dateibasierten Übertragung des Kundenlastenhefts
<i>PTC Integrity Modeler</i>	Entwurf der Systemarchitektur unter Verwendung von SysML	OSLC-RM Consumer zur Repräsentation textbasierter Anforderungen
<i>Sparx Enterprise Architect</i>	Erstellen des Funktionalen Sicherheitskonzepts nach ISO 26262	<ul style="list-style-type: none"> • OSLC-RM Consumer zur Repräsentation der Sicherheitsziele und Update von Sicherheitsfunktionen in der Systemspezifikation • OSLC-Asset Provider zur Kopplung des Modells an das Lifecycle Management Werkzeug
<i>ICOS</i>	Virtuelle Integrationsplattform zur Durchführung von dynamischen Co-Simulationen mit heterogenen Simulationstools	FMI (Functional Mockup Interface) zur dynamischen Kopplung verschiedener domänenspezifischer Simulationsmodelle
<i>AVL VeVaT/Magic</i>	Testplanung und Analyse der Simulationsergebnisse	<ul style="list-style-type: none"> • OSLC-RM Client zur Darstellung der getesteten Anforderungen • OSLC-QM Client zur Aktualisierung von Teststatus und -Ergebnis

Tabelle 1 - Verwendete Tools, Funktionalitäten und Schnittstellen

2.2 Beschreibung des gesamten Szenarios

Ausgangspunkt für das Szenario sind die Anforderungen. Häufig kommen die Anforderungen von einem Kunden (z.B. Automobilhersteller), der ein eigenes Anforderungsmanagement Tool verwendet. Ein Direktzugriff auf die Daten ist meistens

technisch nicht möglich, oder aus Geheimhaltungsgründen unzulässig. Trotzdem ist gerade in der Spezifikationsphase ein iterativer Abstimmungsprozess zwischen Herstellern und Lieferanten erforderlich, um die Anforderungen gemeinsam zu präzisieren bis eine stabile Basis für die weitere Entwicklung geschaffen ist. Der Austauschprozess für Anforderungen der Herstellerinitiative Software (HIS)⁴ dokumentiert diesen Abstimmungsprozess und legt die Verwendung von ReqIF als Austauschformat nahe. Im CRYSTAL Automotive Demonstrator wird dieser Prozess durch die bidirektionale Übertragung der Kundenanforderungen im ReqIF Format zwischen IBM DOORS und dem verwendeten Anforderungsverwaltungs-Werkzeug PTC Integrity LM unterstützt.

Der eigentliche Entwicklungsprozess startet also auf Basis einer mit dem Kunden abgestimmten Anforderungs-Baseline. Um die natürlichsprachlichen, nicht formalisierten Anforderungen nachweisbar zu machen, werden sie mit einem speziellen Tool in eine semi-formale textbasierte Notation gebracht. Weiterhin wird eine modellbasierte Systemspezifikation im SysML Modellierungstool PTC Integrity Modeler aus den Anforderungen abgeleitet. Ein wichtiger Aspekt ist dabei, dass die Anforderungen und die Systemspezifikation konsistent gehalten werden.

Auch bei sicherheitskritischer Entwicklung in Übereinstimmung mit der automotive-spezifischen Norm ISO 26262 wird verstärkt Modellierung eingesetzt. In diesem Szenario werden so genannte Sicherheitsziele (Safety Goals) als spezielle Ausprägung von Systemanforderungen im Modellierungstool eingelesen. Hier wird ein Funktionales Sicherheitskonzept abgeleitet, in dem beispielsweise Sicherheitsfunktionen spezifiziert und Sicherheitsanalysen durchgeführt werden. Die am Ende daraus resultierenden Funktionalen Sicherheitsanforderungen, werden direkt in das Anforderungsverwaltungs-Werkzeug PTC Integrity LM übertragen und dort weiter verwaltet und verlinkt.

Die Validierung der Anforderungen wird im Testwerkzeug AVL VeVaT geplant und gesteuert. Dazu bezieht VeVaT die semi-formalisierten Anforderungen über den OSLC-RM Provider direkt aus PTC Integrity LM und löst eine Simulationdurchführung in ICOS aus. Die Resultate dieser Simulation werden analysiert um schließlich die Informationen über Erfolg (pass) oder Fehlschlag (fail) direkt an den OSLC Provider von PTC Integrity LM zurückzumelden. Auf diese Weise kann jederzeit die Testabdeckung und der Validierungsstatus der Anforderungen gesehen und beispielsweise als Report ausgegeben werden.

3 Teilszenario "Validierung von Anforderungen"

In diesem Artikel soll das Teilszenario „Validierung von Anforderungen“ eingehender betrachtet werden. Abbildung 2 illustriert die Zusammenhänge und der sequenzielle Ablauf des Szenarios ist nachfolgend beschrieben.

⁴ http://portal.automotive-his.de/images/pdf/SimulationandTools/his_ak-rm_exchange_process-1.2_en.pdf

1. Die mit dem Kunden abgestimmte Anforderungsspezifikation des Systems liegt in PTC Integrity LM vor und ist durch eine Baseline identifiziert.
2. Diese nicht formalisierten Anforderungen werden über die OSLC-RM Schnittstelle in ein eclipse-basiertes Werkzeug zur (Semi-)Formalisierung übertragen.
3. In diesem Werkzeug werden die Anforderungen in eine strukturierte, textbasierte Form gebracht.
4. Die resultierenden semi-formalen Anforderungen werden über die OSLC-RM Schnittstelle zurückgeschrieben und mit den ursprünglichen Anforderungen verlinkt.
5. Unter Verwendung der semi-formalen Anforderungen werden Testfälle mit entsprechenden Eingangsparametern definiert und in das Validierungswerkzeug AVL VeVaT übertragen.
6. In der Simulations Plattform ICOS wird die Konfiguration der (Co-)Simulation und der Eingangsparameter vorgenommen.
7. Die Simulation des Fahrzyklus wird unter Verwendung der definierten Validierungsparameter gestartet.
8. Die Simulationsergebnisse werden strukturiert abgelegt und mit dem jeweiligen Testfall verlinkt.
9. Die Simulationsresultate werden in AVL VeVaT analysiert. Hier findet der Vergleich mit den Grenzwerten statt, die in den Anforderungen spezifiziert wurden.
10. Als Ergebnis dieser Analyse werden Validierungsergebnisse als passed/failed Information erstellt und mit weiteren Informationen wie z.B. Grund des Misserfolgs oder Abweichungen von Grenzwerten, angereichert.
11. Diese Validierungsergebnisse werden von AVL VeVaT via OSLC direkt in die Anforderungen in PTC Integrity LM geschrieben.

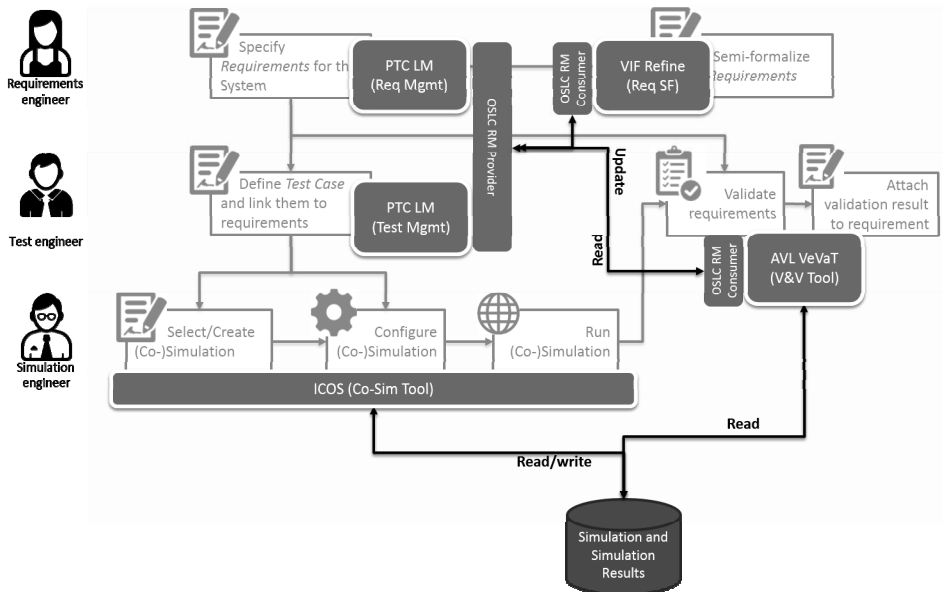


Abbildung 2: Teilszenario "Validierung von Anforderungen"

3.1 Schnittstellen und Verwendung von OSLC

Neben dem Ablauf stellt Abbildung 2 auch die in diesem Szenario verwendeten Tools und deren Schnittstellen dar. Für den Austausch von Anforderungen und deren Verlinkung mit anderen Entwicklungsartefakten wird OSLC RM⁵, die OSLC Spezifikation für Anforderungsverwaltung und -definition (engl: Requirements Management and Definition), verwendet. OSLC ist eine offene Spezifikation für die web-basierte Kopplung von Software Tools im Entwicklungsprozess. Der Fokus liegt dabei nicht auf den Tools oder deren Funktionalität, sondern auf der Integration der Daten. OSLC baut auf bekannten und etablierten Webtechnologien auf, wie HTTP und RDF. Basierend auf der Core Spezifikation, die allgemeine Grundlagen definiert, gibt es die so genannten Domänen-Spezifikationen. Diese legen das Vokabular und die so genannten Ressourcen – also Datenobjekte – in einem konkreten Anwendungsbereich fest, wie z.B. Requirements Management, Quality Management, Architecture Management, usw.

Eine OSLC Toolkopplung wird dadurch ermöglicht, dass zwei Tools prinzipiell zwei unterschiedliche Rollen implementieren: Der „Provider“ stellt die Daten als OSLC Ressourcen bereit und der „Consumer“ greift auf diese lesend oder auch schreibend zu. Über die Dienste des Providers können auch neue OSLC Ressourcen angelegt, oder bestehende gelöscht werden. Jedes Tool kann gleichzeitig unterschiedliche Rollen implementieren und so beispielsweise gleichzeitig ein Provider für Anforderungen und ein Consumer von Testfällen sein. Aufgrund der Verwendung eines gemeinsamen Vokabulars ist es für die Toolkopplung irrelevant wie die Daten toolintern dargestellt werden, solange die gemeinsame standardisierte Repräsentation nach außen erhalten bleibt. Grundsätzlich gilt bei der Spezifikation der Interfaces: Soviel wie notwendig, aber so wenig wie möglich.

Die Functional Mock-up Interface Spezifikation (FMI⁶) unterstützt sowohl den Austausch von Modellen als auch die Durchführung von Co-Simulationen. Das Ziel ist die Kopplung von zwei oder mehreren Simulationstools zu einer Co-Simulation von beispielsweise dem Verhalten eines Gesamtsystems. Der Datenaustausch ist dabei auf diskrete Synchronisationspunkte beschränkt. Zwischen diesen Synchronisationspunkten rechnen die Simulationstools unabhängig voneinander. Das hier verwendete Co-Simulationstool ICOS unterstützt den FMI Standard und fungiert in der Co-Simulation als Mastertool, das den Austausch der Signalwerte zu den definierten Synchronisationspunkten verwaltet und koordiniert, mit dem Ziel den durch die Co-Simulation entstehenden Extrapolationsfehler zu minimieren.

⁵ <http://open-services.net/wiki/requirements-management/>

⁶ <https://www.fmi-standard.org/>