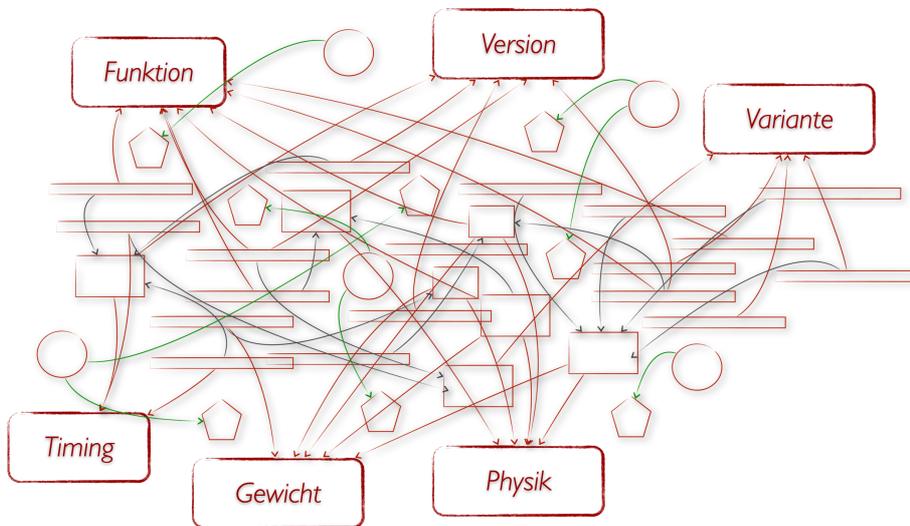


Von Dokumenten zu Linked Data

Voraussetzung zum effizienten Engineering von Komplexen Systemen

Embedded Software Engineering Report Nr. 42



Komplexe Systeme verstehen und managen auf Basis von Beziehungen (Relationen)

Wenn es ein Projekt gibt mit extremer Komplexität, entstanden mit dem Ziel Zusammenhänge bewusst zu verschleiern, um Verstehbarkeit zu verhindern, dann zeigt es sich bei der Analyse der ‚Panama Papers‘.

5 Mio E-Mails, 3 Mio Datenbank-Einträge (in verschiedensten Formaten), 2 Mio PDF-Dokumente, 1 Mio Bilder und 0,5 Mio Text-Dokumente.

Um in diesen ‚Papers‘ Informationen, Strukturen und Zusammenhänge zu erkennen, wurde ein neues Paradigma für die Analyse angewandt. Sehr simplifiziert dargestellt wurde dabei wie folgt vorgegangen:

- In einem ersten Schritt wurden alle Informationen analysiert und Entitäten (wir sprechen in Folge von Artefakten) definiert.
- In einem zweiten Schritt wurden Zusammenhänge und Relationen zwischen diesen Entitäten analysiert und diese auf Basis einer Graph-orientierten Datenbank verlinkt. (Linked Data).
- In einem dritten Schritt konnten auf Basis von Linked Data neue Sichten erstellt werden, die Zusammenhänge erkennbar gemacht und damit Verständnis ermöglicht haben.

(Das war jetzt extrem vereinfacht, genauer hier nachzulesen: <https://neo4j.com/blog/analyzing-panama-papers-neo4j/>)

Analog kristallisiert sich die Möglichkeit, dynamisch Viewpoints auf Basis von Linked Data über alle Engineering Disziplinen hinweg zu generieren, zunehmend als neues Paradigma heraus, um komplexe Systeme zu verstehen und zu engineer.

Um diesen Ansatz zur Strukturierung von Engineering Artefakten soll es in diesem ESER gehen.

Inhalt:

Komplexe Systeme verstehen und managen auf Basis von Relationen

Strukturierung von Engineering Informationen

Das Dokumenten-zentrische Paradigma

Das neue Paradigma: Beziehungsstrukturen anstelle von Containern

Voraussetzungen

Praktische Lösungen, OSLC und Global Configuration

Wiederverwendung von Artefakten unabhängig von Link - Beziehungen

Anforderungen an die IT Infrastruktur

Resümee

Vertiefende Themen

Strukturierung von Engineering Informationen

Beim Engineering können grundsätzlich zwei Bereiche differenziert werden:

1. Die Engineering Prinzipien, die unmittelbar beim Engineering der Produkte oder Systeme angewendet werden.
2. Die Ressourcen und mittelbaren Prinzipien, die ein effizientes Engineering ermöglichen, wie zum Beispiel der Arbeitsplatz mit Schreibtisch und Rechner oder eine IT Infrastruktur. Zu dieser grundlegenden Infrastruktur gehört auch die Metastruktur, in der Engineering Artefakte abgelegt und strukturiert werden.

Diese Metastruktur ergibt sich im wesentlichen aus den angewandten Prozessen und der sich daraus ableitenden Abfolge der Arbeitsschritte. Historisch gesehen wurden, in Anlehnung an das V-Modell, nacheinander die Phasen Analyse und Spezifikation, Design und Implementation und am Ende Verifikation durchlaufen. Lange Zeit galt die Regel, dass eine neue Phase erst dann beginnen kann, wenn die vorherige vollständig abgeschlossen ist. Mit der Implementation wird also erst begonnen, wenn die Spezifikation abgeschlossen ist.

Vor diesem Hintergrund ist es naheliegend, Engineering Artefakte in Phasen-bezogenen Containern zu strukturieren und zu speichern: alle Spezifikations-Artefakte liegen in einem Word-Dokument; alle Programmierungs-Artefakte liegen in C-Modulen, etc..

Heute gerät dieses starre, Phasen-bezogene Konzept, Artefakte zu strukturieren zunehmend an seine Grenzen. Dazu tragen ganz wesentlich der Trend zu agilen Methoden sowie die kontinuierlich zunehmende Komplexität der zu entwickelnden Systeme bei.

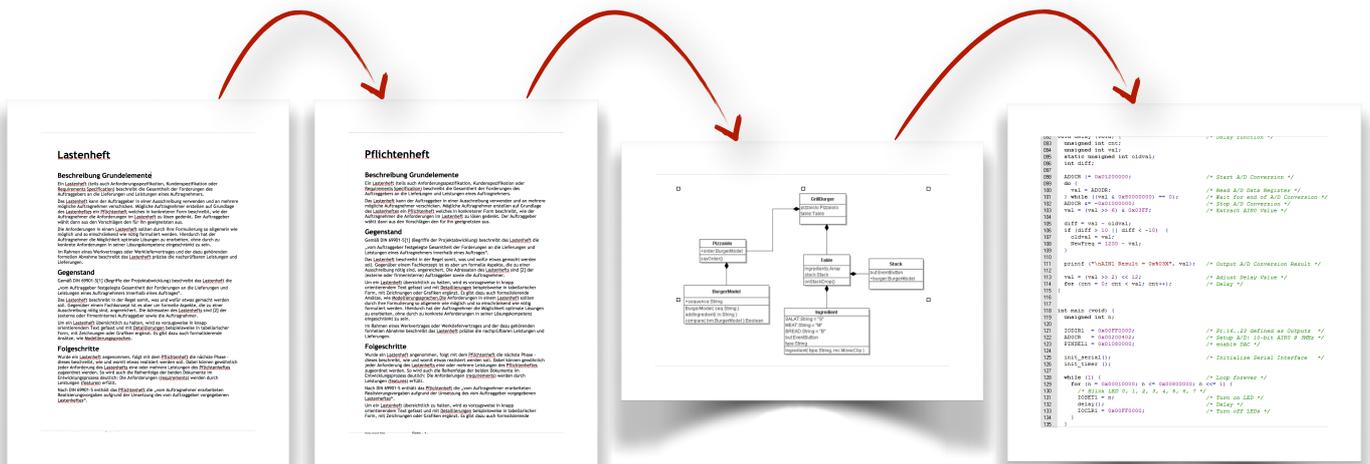


Abb. Nr. 1: Dokumente als Artefakte

Das Dokumenten-zentrische Paradigma

Zwei wesentliche Aspekte unterscheiden das Engineering in der Vergangenheit vom heutigen Engineering.

1. Die Annahme, dass es ideal ist, die Engineering Schritte in Phasen zu durchlaufen, die abgeschlossen sind, bevor mit einer neuen Phase begonnen wird. (Dieses Vorgehen ist dadurch geprägt, dass ein größerer Anteil an Produkten oder Systemen von Grund auf neu entwickelt wurden. Heute werden in der Regel existierende Systeme um Funktionen erweitert und/oder neu zusammengesetzt.)
2. Eine vermeintlich einfache eindimensionale Sicht auf das Engineering war ausreichend.

Orientiert an diesem Vorgehen basiert die Struktur der Engineering-Artefakte auf einzelnen Containern, in denen jeweils die Artefakte einer Phase zusammen abgelegt sind. Alle Artefakte, die für die Systemspezifikation relevant sind, liegen in einem Container in Form eines Dokumentes; alle Implementierungs-Artefakte liegen in Containern in Form von C-Modulen; Verifikations-Artefakte liegen vielleicht wieder in einem Dokument, etc..

Ein Nachteil dieses Ansatzes ist, dass sich ein einzelnes Engineering-Artefakt immer nur in einem Container gemeinsam mit allen anderen Artefakten derselben Phase befinden kann. Die Struktur ist eindimensional und erlaubt nur eine lineare Sicht. Container aufzuteilen in Subcontainer und die Verwendung von Kapitelstrukturen innerhalb der Container ermöglichen weitere Perspektiven. Aber diese verfeinern lediglich und können nicht

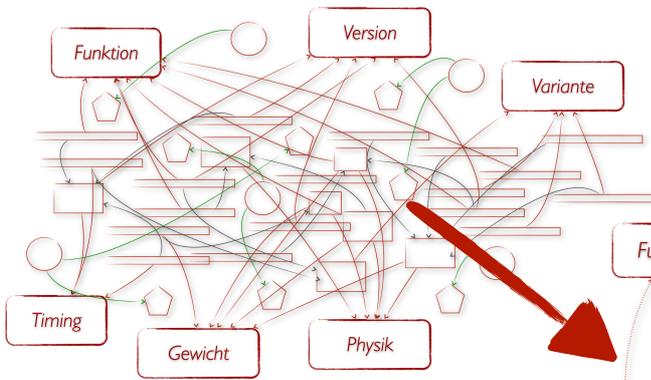
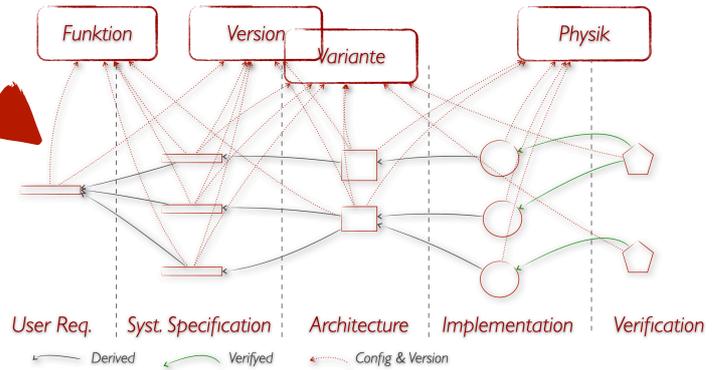


Abb. Nr. 3: Viewpoint erzeugt aus Beziehungsdatenbank



Im Gegensatz zum eingangs beschriebenen Konzept der isolierten Container kann ein individuell adressierbares Artefakt unbegrenzt viele Link-Beziehungen aus verschiedensten Gesichtspunkten zu anderen Artefakten haben.

Für jeden Gesichtspunkt gibt es eine spezifische Link-Beziehung. Auf diese Weise lassen sich vielfältige Gesichtspunkte zum einen unabhängig voneinander aber auch in ihren Beziehungen zueinander auf Basis von ‚Viewpoints‘ darstellen.

So können zum Beispiel über die Link-Beziehung ‚Derived‘ / ‚Satisfies‘ die Beziehungen aller Artefakte, die funktionale Zusammenhänge haben, über die Prozessschritte hinweg dargestellt werden.

In der Link-Beziehung ‚Baseline‘ können alle Artefakte in Beziehung zu einer bestimmten Baseline dargestellt werden.

Question: If this artifact changes – what is the impact up/down stream ?

answer

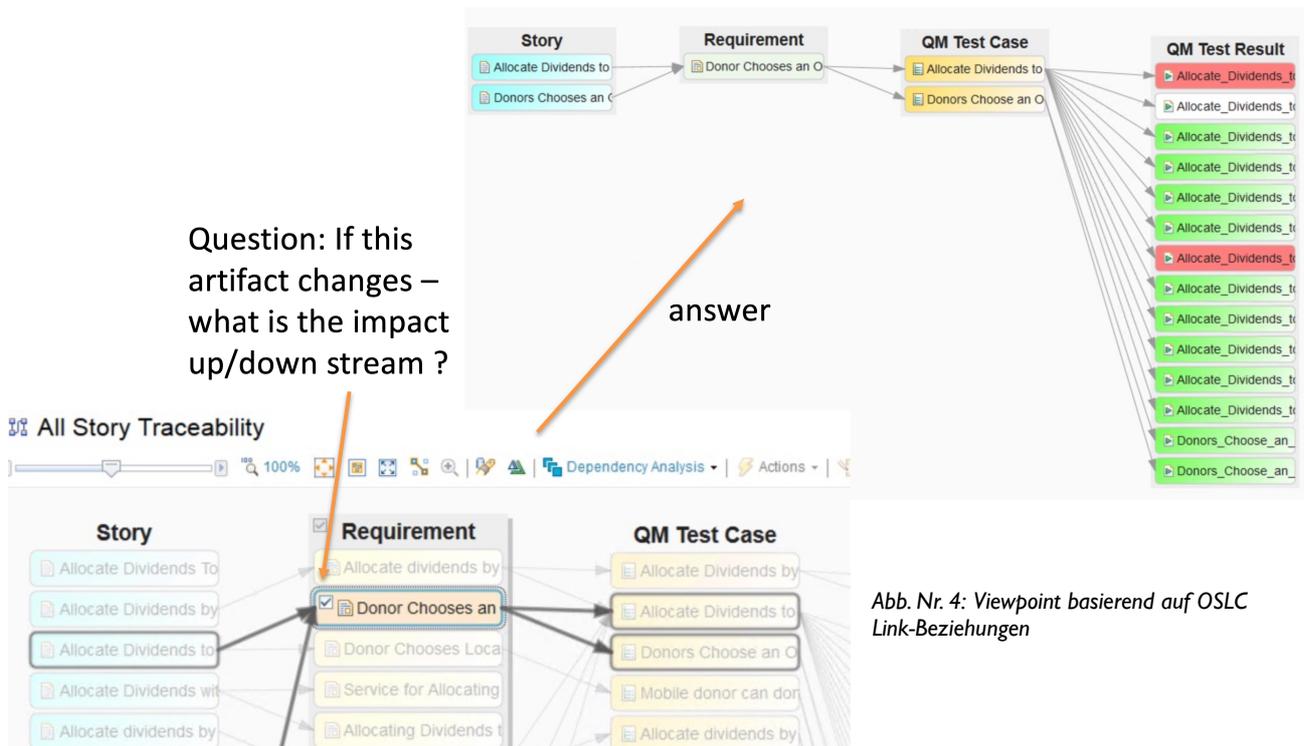


Abb. Nr. 4: Viewpoint basierend auf OSLC Link-Beziehungen

In der Link-Beziehung ‚Changeset‘ können alle Artefakte in Beziehung gesetzt werden, die im Kontext einer funktionalen Änderung berührt wurden, zum Beispiel für die Übergabe zum Review.

Diese Sichten können sich auch gegenseitig überschneiden. So kann zum Beispiel ein Artefakt in einer bestimmten Version gleichzeitig einer bestimmten Variante zugeordnet werden. Das selbe Artefakt kann gleichzeitig in einem Change Set liegen und/oder in einer anderen Variante.

Der Raum der Zuordnung und der daraus bildbaren Viewpoints ist (zumindest theoretisch) unendlich. Das gilt ebenso für daraus ableitbare Arbeitsschritte und/oder eventuelle Automatisierungen.

Voraussetzungen

Wie weiter oben beschrieben, sind es zwei Bedingungen, die für den Übergang vom Dokumenten-zentrischen zum relationalen Paradigma zwingend erfüllt sein müssen:

1. Die Speicherung von Linkbeziehungen zwischen Artefakten über Engineering-Disziplinen und Tools hinweg.
2. Die eindeutige Adressierbarkeit der einzelnen Artefakte. Dies ist in den Dokumenten-zentrischen Strukturen in der Regel nicht vorgesehen und benötigt Datenbanken (*Repositories*), in denen einzelne Artefakte abgelegt werden und mit UID's eindeutig adressierbar sind.

Letztendlich benötigt es zudem Traceability-Analysen und daraus generierte Viewpoints.

Unique Identifier ,UID‘

(dt.: eindeutige Identifizierungsnummer ,UIN‘) ist eine Ziffern- oder Zeichenkombination, mit der sich Hardware und Software, Personen, Kommunikation, Gegenstände oder allgemein Entitäten innerhalb einer bestimmten Kategorie eindeutig identifizieren lassen.

Quelle: <https://www.xovi.de/was-ist-ein-unique-identifier>

Praktische Lösungen, OSLC und Global Configuration

Die Anwendung des relationalen Paradigmas findet im Kontext des Anforderungs-Management mit Werkzeugen wie IBM DOORS, Siemens Polarion oder IBM DOORS Next Gen bereits statt. Dort wird heute

CR-5	2 Detailed Description		
CR-6	2.1 Quality of sound		
CR-7	The digital to analogue conversion has to be based on a sample-frequency of 44,1kHz and 16bit resolution.	SS-45 The digital to analogue conversion has to be based on a sample-frequency of 44,1kHz and 16bit resolution.	ASGR-5 The Timing is based on the Samplefrequency 44.1 kHz. ASGR-6 The resolution is based on 16 Bit. ATR-5 Timing of the Timer is based on the Sample Frequency of 44.1 kHz
CR-8	The soundsystem has to be able to produce at least 5Watts with a maximum distortion factor of 1%.	SS-46 The soundsystem has to be able to produce at least 5Watts with a maximum distortion factor of 1%.	
CR-9	2.2 Timbre		
CR-10	The soundsystem has to support only one timbre and should be close to the sound of an Piano	SS-48 The soundsystem has to support only one timbre and should be close to the sound of an Electronic Organ. SS-49 The frequency-response is from 20Hz to 18kHz. SS-50 Polyphony up to 20 simultaneous tones.	NCR-4 Calculates Sine-values for different sound Timbres. At the moment one sound (Electronic Organ) is implemented. ASGR-5 The Timing is based on the Samplefrequency 44.1 kHz. ASGR-3 The AudioSoundGen has to read the valid notes from the Array provided by the NoteController and has to program the DA Converter. NCR-3 For polyphonic Sound this controller provides an Array of sine-values for the AudioSoundGen.

Abb. Nr. 5: Traceability View, Abhängigkeit von Anforderungen.

schon mit Link-Beziehungen gearbeitet, die Traceability-Analysen ermöglichen.

Diese Fähigkeit ist übrigens eines der wichtigsten Argumente für die Nutzung von Requirements Engineering-Werkzeugen im Vergleich zu MS Word oder MS Excel. Die neue Herausforderung liegt darin, Link-Beziehungen und darauf basierende Traceability-Analysen über Engineering-Disziplinen hinweg zu ermöglichen.

Es gibt Lösungen, die Artefakte zwischen Werkzeugen kopieren (*Synchronisieren*) und auf dieser Basis dann Links innerhalb eines Werkzeuges ermöglichen. Dieses Vorgehen gestaltet sich aufwendig und fehleranfällig und skaliert nicht für große Projekte.

Seit etwa 2013 gibt es den OSLC Standard (Open Services for Lifecycle Collaboration). Er definiert eine ,Service Orientierte Architektur‘, in der Link-Beziehungen eine der Grundlagen bilden. Auf Basis dieses Standards ist es möglich, Link-Beziehungen über Domänengrenzen und Werkzeuge hinweg zu erstellen. Mit Hilfe von Traceability-Analysen können dann auch Viewpoints erstellt werden, die unterschiedlichste Sichten über das gesamte System ermöglichen und damit auch Zusammenhänge aufzeigen, die Domänen-inhärent nicht zum Ausdruck gebracht werden können.

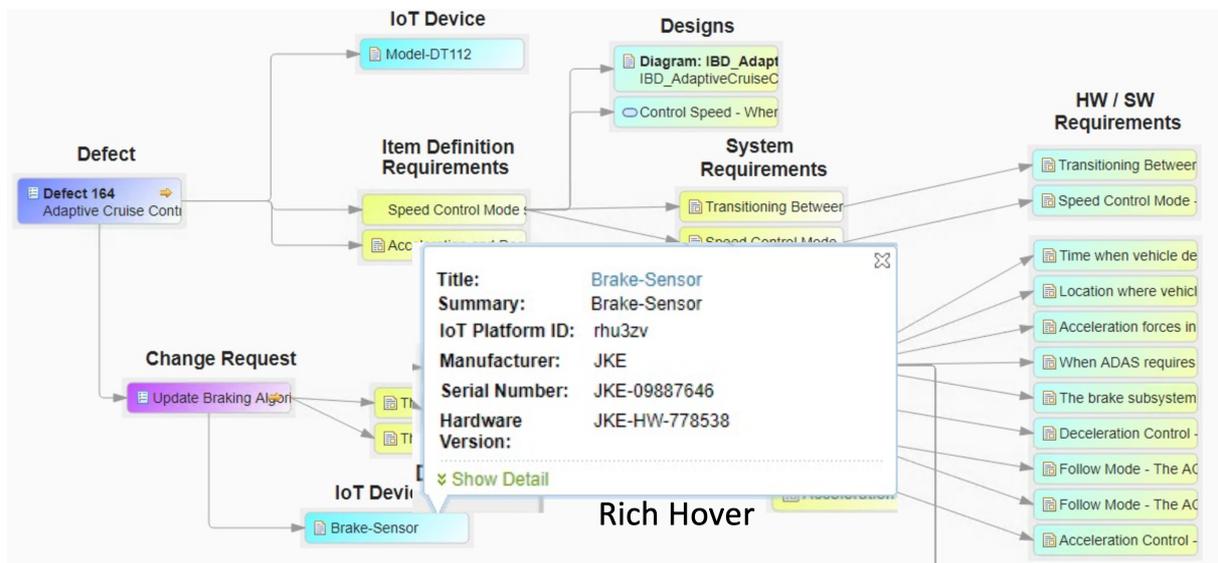


Abb. Nr. 6: Domänen-übergreifender Viewpoint zum Beispiel auch auf Produktionsdaten mit Live Data und Rich Hover

Diese Viewpoints basieren auf ‚Live Data‘ und unterstützen auch ‚Rich Hover‘. Das bedeutet, innerhalb der Sicht aus einem Werkzeug (Repository) werden auch Daten aus anderen Werkzeugen live angezeigt. Mit dem Bewegen (Hovern) der Maus über ein Artefakt aus einem anderen Werkzeug werden weitere Daten bis hin zu Diagrammen aus dem anderen Repository live abgerufen und angezeigt.

Wiederverwendung von Artefakten unabhängig von Link-Beziehungen

Oben wurde gesagt, dass das Paradigma der Link-Beziehungen die unbegrenzte Darstellung von Sichten ermöglicht. Das war theoretisch und idealisiert. In der Praxis gibt es die eine oder andere Einschränkung.

Eine relevante Einschränkung in Bezug auf die Abbildung von Varianten, die sehr häufig in der Praxis existiert, resultiert aus der Möglichkeit der Wiederverwendung von Artefakten losgelöst von Link-Beziehungen. Das ist notwendig, um die Traceability zum Beispiel für spezifische Varianten darzustellen.

In den meisten Werkzeugen ist das nicht möglich, was bedeutet, dass ein Artefakt alle Link-Beziehungen über alle Varianten hinweg aufsummiert.

Für die Traceability in Bezug zu einer spezifischen Variante ist es aber sehr sinnvoll und hilfreich, wenn die Link-Beziehungen gefiltert auf diese Variante dargestellt werden können. Oder umgekehrt ausgedrückt ist die Einschränkung dieser Möglichkeit so einschneidend, dass Varianten in diesen Werkzeugen dann nicht auf Basis von Link-Beziehungen, sondern auf Basis von Attributen abgebildet werden müssen. Das führt in der Praxis zu Mehraufwand und technischen Einschränkungen.

Auf Basis von Relationen kann die Kombination aus Versionen und Varianten sehr effizient abgebildet und gehandhabt werden. Dafür ist es jedoch notwendig, dass dieselben Artefakte in verschiedenen Varianten unterschiedliche Link-Beziehungen haben können. Das unterstützen leider aktuell nur sehr wenige Werkzeuge. Bei der Auswahl eines Werkzeuges sollte diese Fähigkeit berücksichtigt werden.

Anforderungen an die IT Infrastruktur

Link-Beziehungen und Traceability-Analysen innerhalb eines Werkzeuges sind heute häufig Stand der Praxis und stellen erst im Kontext von großen Projekten besondere Anforderungen an die IT. Im Kontext von Domänen- und Werkzeug-übergreifendem Management sind Link-Beziehungen und Traceability-Analysen als ‚Live Data‘ eine ganz andere Herausforderung. Sie stellt neue Anforderungen nicht nur an die benötigte IT Infrastruktur sondern vor allem auch an die Skalierbarkeit der eingesetzten Werkzeuge.

Die Zauberworte heißen hier: vertikale und horizontale Skalierbarkeit. Moderne, Repository-basierte Engineering-Werkzeuge verwenden häufig professionelle Datenbanksysteme und zugehörige Anwendungsserver. Vertikale Skalierbarkeit bedeutet die Möglichkeit, existierende Werkzeuge durch performantere Hardware (mehr/schnellere CPUs, mehr RAM, mehr und schnellerer Storage) zu beschleunigen. In der Regel haben Werkzeuge klare Obergrenzen, bis zu welcher Repository-Größe und -Komplexität sie

durch „mehr direkte Ressourcen“ wachsen können - sogenannte ‚upper limits‘, die vom Hersteller wohl dokumentiert sein sollten.

Hier kommt die horizontale Skalierbarkeit ins Spiel: gemeint ist die Fähigkeit eines Werkzeuges, durch Hinzufügen von weiteren physikalischen oder virtuellen Maschinen die Gesamtperformance zu erhöhen - sei es auf Ebene der Datenbank oder der Anwendungsserver. Ein Stichwort ist ‚Clustering‘ und ‚Application Load Balancing‘. Wenige Werkzeuge am Markt unterstützen zur Zeit vollständiges Clustering, also die komplette, gleichmäßige Verteilung der Rechenlast auf beliebig viele Systeme. Üblicher ist es, ab einer gewissen Repository-Größe einzelne Engineering-Disziplinen oder gar Teilprojekte einer einzelnen Disziplin auf eigenständige Datenbank- und/oder Applikationsserver zu verteilen. Dies sollte ein Werkzeug unter Beibehaltung der vorhandenen Metastruktur unterstützen.

Wie eine hochperformante IT Infrastruktur für komplexe Engineering-Werkzeuge aussehen kann, haben wir in einem Dokument dargestellt: ["An operational concept for a vertically and horizontally scalable IT infrastructure"](#)
Bei Interesse senden wir es Ihnen gerne zu. [Anfrage an: swillert@willert.de](mailto:swillert@willert.de)

Resümee

Der Übergang von einem Dokumenten-zentrischen zum Relations-zentrischen Paradigma ermöglicht unbegrenzte Freiräume in der Darstellung von Engineering-Aspekten und -Sichten, auch wenn diese orthogonal zueinander stehen. Das ist eine Notwendigkeit für das Engineering von komplexen Systemen im Kontext von Versionen und Varianten.

Beim Übergang zum neuen Paradigma kann es schwer fallen, die Orientierung zu behalten. Dies gilt insbesondere, wenn gewohnte Strukturen wegfallen, zum Beispiel die Kapitelstruktur innerhalb eines Dokumentes.

Moderne gute Werkzeuge ermöglichen trotz Artefakt und Beziehungs-zentrischen Strukturen auch die Nachbildung von Dokumenten mit Kapitelstrukturen als möglichen ViewPoint. Das kann für einige Sichten weiterhin sehr sinnvoll und hilfreich sein. Damit hat die neue Generation von Werkzeugen und Prinzipien mit signifikanten Vorteilen fast keine Nachteile gegenüber den alten.

Letztendlich bleiben lediglich deutlich höhere Anforderungen an die IT Infrastruktur, aber auch diese halten sich gegenüber den Personalkosten in Grenzen.

Engineering-Disziplinen, Phasen und Werkzeug-übergreifende Strukturierung von Engineering-Artefakten und die Bildung von entsprechenden Viewpoints werden im Engineering-Kontext von Komplexen Systemen immer wichtiger. Große Bereiche dieses Gesichtspunktes werden automatisch bereitgestellt, wenn die Werkzeuge OSLC und Global Configuration unterstützen.

Aber Achtung: Häufig versprechen Marketing-Aussagen einiger Tool-Hersteller mehr als die Tools in der Praxis liefern können.

Vertiefende Themen:

Analyse der 'Panama Papers' auf Basis des ‚Nodes and Relationship‘ Ansatzes

<https://neo4j.com/blog/analyzing-panama-papers-neo4j/>

SECollab

Eine Graph-Datenbank-Lösung, basierend auf Neo4J zur Analyse, Review und Dokumentation von Engineering-Artefakten. SECollab unterstützt OSLC und hat verschiedene Schnittstellen zu Engineering Tools, die kein OSLC unterstützen: <https://www.sodiuswillert.com/en/products/secollab>

OSLC (Open Services for Lifecycle Collaboration)

Standard REST APIs to connect data: <https://open-services.net/>

Erklärvideo: www.youtube.com/watch?v=Nuc63RGpXII&t=4s

Technische Voraussetzungen für eine OSLC IT Infrastruktur

"An operational concept for a vertically and horizontally scalable IT infrastructure"

Bei Interesse senden wir Ihnen das Dokument gerne zu. [Anfrage an: swillert@sodiuswillert.com](mailto:swillert@sodiuswillert.com)

Weitere Ausgaben von:

EMBEDDED SOFTWARE ENGINEERING - REPORT

finden sie unter:

www.willert.de/Newsletter

Technical deep dive:

<https://rhapsody.blog>

www.willert.de/techletter

SodiusWillert Blogs:

www.sodiuswillert.com/de/blog

www.sodiuswillert.com/en/blog

Autor:

ANDREAS WILLERT

Herausgeber:

**WILLERT SOFTWARE TOOLS
GMBH**

Hannoversche Straße 21

31675 Bückeberg

info@sodiuswillert.com

Tel.: +49 5722 9678 - 60

sodiuswillert.com/de

Willert Software Tools ist Teil der Unternehmensgruppe

