

Integration von Systemmodellen mit fünf fundamentalen Elementtypen

Oskar von Dungern

enso managers gmbh, Charlottenstrasse 68, 10117 Berlin, od@enso-managers.com

Zusammenfassung: Bei der Konzeption und Spezifikation mechatronischer Systeme entstehen Teilergebnisse durch verschiedene Beteiligte, die sich unterschiedlicher Methoden und Werkzeuge bedienen. Es besteht ein großes Interesse, diese Informationen in einen gemeinsamen Kontext zu stellen und inhaltlich zu integrieren. Dazu wird ein übergreifendes Systemmodell gebildet, welches interessierende Aspekte der einzelnen Teilergebnisse zusammen fasst.

Indem identische Modellelemente verschiedener Diagrammtypen/Methoden konsolidiert und logische Beziehungen zwischen Modellelementen aufgebaut werden, wird die Konsistenz verbessert und der Gehalt an Bedeutung erhöht. Konsolidierung und Vernetzung werden erheblich vereinfacht, wenn die originalen Modellelemente abstrahiert und auf die fundamentalen Modellelemente Ereignis, Akteur, Zustand, Anforderung und Merkmal abgebildet werden.

Eine Formalisierung mit Augenmaß reduziert den Interpretationsspielraum und verbessert die Vermittlung der Inhalte. Das resultierende semantische Netz des integrierten Systemmodells kann (teilweise) automatisch verifiziert und die Qualität des übergreifenden Systemkonzepts besser bewertet werden, bevor die Entwicklung in den Disziplinen Mechanik, Elektronik und Software beginnt.

1. Aufgabenstellung

Aus inhaltlicher Sicht stammen eingehende Informationen meist aus verschiedenen Teilorganisationen und beleuchten Anforderungen aus Gesetzen, Verbraucherschutz und Produktstrategie, parametrische Auslegungen oder ähnliche Informationen. Um Informationen und Modelle verschiedener Quellen zusammenführen zu können, ist ein gemeinsamer Kontext zu schaffen. Dabei werden folgende Ziele verfolgt:

- Beim Erstellen von Systemspezifikationen methodische Unterstützung erfahren,
- In Informationen verschiedener Quellen einheitlich suchen und navigieren,
- Identische Elemente in verschiedenen Modellen finden und konsolidieren,
- Abhängigkeiten und logische Beziehungen zwischen Modellelementen hinterlegen, also verschiedene Modelle semantisch verknüpfen,
- Qualität verbessern durch früheres Erkennen von Fehlern oder Verletzungen von Entwurfsregeln,
- die interdisziplinäre Zusammenarbeit bei der Systemkonzeption unterstützen.

Nachfolgend wird ein Verfahren zur Diskussion gestellt, das hierfür die Voraussetzungen schafft.

2. Definitionen

In aller Kürze werden einige Begriffe abgestützt auf Wikipedia [Wik15] erläutert, wie sie nachfolgend verstanden werden:

- Modell:* Abbild der Wirklichkeit, das bestimmte Aspekte heraushebt und mit zweckentsprechender Genauigkeit wiedergibt.
- Sprache:* Vokabular (oder Notation) sowie Regeln zur Verständigung in einer Sprachgemeinschaft, darunter v.a. Syntax und Semantik.
- Vokabular:* Gesamtheit der Bedeutung tragenden Wörter einer Sprache.
- Notation:* Menge symbolischer Zeichen mit vereinbarter Bedeutung.
- Syntax:* Zusammenfügungsregeln für Wörter; sie bestimmen den Satzbau.
- Semantik:* Interpretationsregeln für Sätze; sie bestimmen die Bedeutung.
- Format:* Codierungsart von Daten.
- Medium:* Kommunikationsmittel bzw. Informationsträger.

3. Vom Auge zum Bit

Das vorgestellte Verfahren zur Modellintegration beruht auf der Repräsentation und Struktur von Information eines Systemkonzepts. In diesem Kapitel werden vier Ebenen identifiziert, die die gleiche Information unterschiedlich darstellen; zwischen ihnen findet jeweils eine Transformation statt. Es wird mit der vom Auge wahrgenommenen Information begonnen und der Weg bis zum atomaren Datenelement im digitalen Speicher nachvollzogen. Die Organisation der Datenspeicherung bestimmt maßgeblich, welche maschinelle Unterstützung beim Erstellen, Suchen und Prüfen der Inhalte möglich ist und wie effizient sie realisiert werden kann.

Medium und Format

Der Bearbeiter erstellt oder liest Systemspezifikationen je nach Anforderungen des Projekts und den Festlegungen der Organisation in unterschiedlichen Medien und Formaten. Verbreitet sind Bürodokumente aus Textverarbeitung oder Tabellenkalkulation einerseits oder dynamisch aufgebaute online-Dokumente in Content-Management-Systemen, Wikis oder vergleichbaren Systemen andererseits. Charakteristisch für beide Fälle ist, dass der Autor die sichtbaren Gliederungen, Texte und Diagramme direkt erzeugt und so ihre Konsistenz, Vollständigkeit und Verständlichkeit, d.h. ihre Qualität alleine bestimmt. Das System kann weder formale noch inhaltliche Unterstützung bieten.



Fig. 1: Systemspezifikation in diversen Medien/Formaten; hier Dokument als Broschüre und am Bildschirm

Notation

Die Verständigung zwischen den Beteiligten wird verbessert, wenn beispielsweise Anforderungen mittels Satzschablonen eindeutiger formuliert und komplexere Zusammenhänge durch Diagramme mit vereinbarter Notation dargestellt werden [Poh11]. Zeichenwerkzeuge bieten vorgefertigte Symbole für die gewählte Notation an und beschleunigen das Erstellen von Diagrammen, doch der Autor erfährt noch immer keine inhaltliche Unterstützung bei ihrer Anwendung.

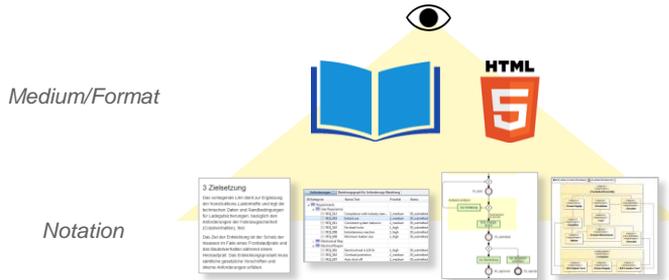


Fig. 2: Vereinbarte Notationen unterstützen die Verständigung, hier ‚Freitext‘, ‚Anforderungstabelle‘, BPMN Prozess, SysML ‚Internal Block Diagram‘.

Modell

Bei der Modellierung werden nicht nur Diagramme erzeugt, sondern zu jedem Diagrammelement gehört auch ein entsprechendes Datenelement, das weitere Information trägt. Diagrammelemente visualisieren also bestimmte Modellelemente, wobei mehrere Visualisierungen des selben Elements auf ein Datenelement verweisen. Änderungen etwa des Namens wirken sich auf alle Visualisierungen aus. Der Nutzer wird unterstützt, indem die Anordnung der Diagrammelemente den Syntax-Regeln der jeweiligen Methode unterworfen wird; es sind nur noch ‚korrekte‘ Anordnungen zulässig.

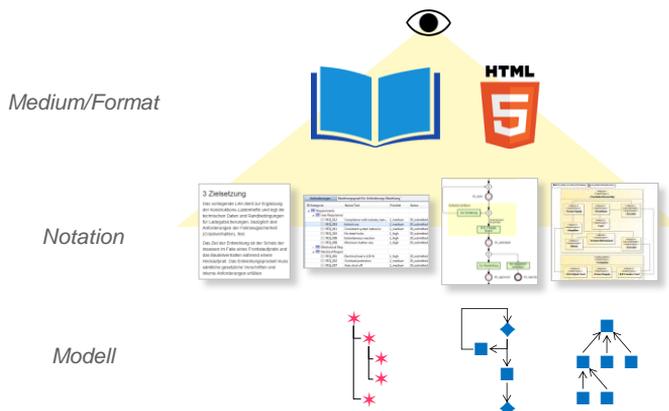


Fig. 3: Diagramme visualisieren logische Modelle

Bei einigen Modellierungs-Werkzeugen wird aus der *Anordnung der Diagrammelemente*, etwa nachdem, ob sie ineinander gezeichnet oder durch eine Kante verbunden sind, eine *logische Bedeutung* abgeleitet und durch explizite Beziehungen der Modellelemente gespeichert. Anders als reine Diagramme lassen vernetzte Modellelemente eine effiziente Suche zu, etwa zur Auflistung aller Teilkomponenten der Komponente ‚Spindel‘ oder aller Funktionen, die das Datenelement ‚Spindeldrehzahl‘ lesen. Weiterhin lässt sich automatisch überprüfen, ob die Vielzahl manuell erstellter Diagramme konsistent ist; beispielsweise kann ‚Spindel-Lager‘ nicht zugleich Bestandteil von ‚Spindel‘ sein und umgekehrt.

Zumeist befasst sich der Standard einer Modellierungsmethode jedoch *nicht* mit der Bedeutung hinter Diagrammen und Modellelementen. Das gilt auch für die Unified Modelling Language (UML) und der darauf aufgebauten System Modelling Language (SysML). Genau genommen handelt es sich bei diesen Methoden nicht um eine Sprache, sondern um eine Notation mit Syntax. Eine Bedeutung wird zwar implizit nahegelegt, doch konkret den Werkzeugh Herstellern oder den Nutzern überlassen. Ohne Festlegung der Interpretationsregeln, der Semantik, verbleibt ein Interpretationsspielraum, der weiterer Erläuterung bedarf. Eine automatische Prüfung verbietet sich durch die verbleibende logische Unschärfe.

Integriertes Modell

Natürlich ist es unbefriedigend, wenn es für die gleiche Komponente mehrere Modellelemente in den Datenbeständen verschiedener Werkzeuge gibt, wie es heute noch meist der Fall ist. Daraus ergibt sich das Ziel alle Modellelemente samt ihrer logischen Beziehungen in einen gemeinsamen Kontext zu überführen, mehrere Instanzen der gleichen Komponente zu konsolidieren und ggf. weitere Beziehungen zu ermitteln und zu hinterlegen. Dann beziehen sich alle Diagramme aller Quellen auf gemeinsam genutzte Elemente im Integrationsmodell (Fig. 4).

Unabhängig davon, ob ein zentraler oder ein föderativer Ansatz der Modellintegration gewählt wird, ist ein Weg zu finden mit der hohen Anzahl von Modellelement-Typen umzugehen: Je mehr Typen es gibt, desto schwerer ist die Zuordnung. Nicht nur ist ihre Bedeutung in verschiedenen Methoden unterschiedlich, sondern auch innerhalb einer Methode gibt es nicht selten eine große Anzahl von Elementtypen. Beispielsweise unterscheidet der SysML Standard 163 graphische Knotentypen, 22 Datentypen, 211 Metaklassen und 25 sog. Stereotypen [Sml12]. Die Autoren kommentieren: “The use of more formal constraints and semantics may be applied in future versions to further increase the precision of the language” [Sml12, S.19].

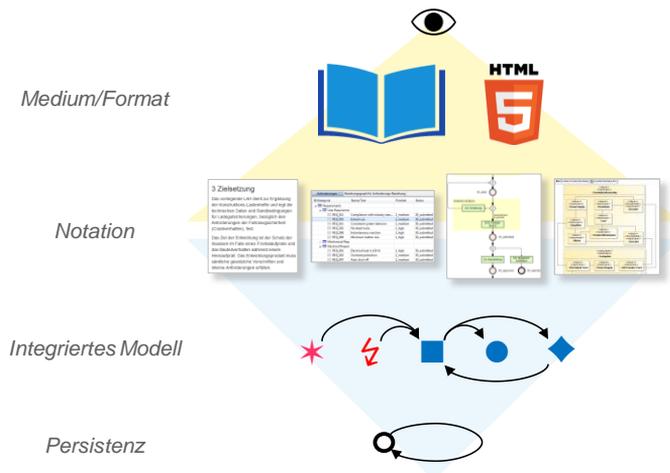


Fig. 4: Diagramme sind Sichten eines integrierten Systemmodells bestehend aus Modellelementen und ihren Beziehungen

Eine derart umfangreiche Notation lässt sicherlich eine besonders nuancierte Ausdrucksweise zu, doch eine Integration oder gar eine Prüfung wird sehr erschwert. Im nachfolgenden Kapitel wird ein gangbarer Weg zur Diskussion gestellt, wie das zuvor postulierte Ziel, nämlich alle Modellelemente samt ihrer logischen Beziehungen in einen gemeinsamen Kontext zu überführen, erreicht werden kann.

4. Modellintegration

Wie ist nun praktisch eine übergreifende Systemkonzeption und eine semantische Vernetzung der Inhalte zu erreichen? Nach unserer Erfahrung ist ein Modellkern unter Verwendung weniger, klar unterscheidbarer Modellelement-Typen erheblich einfacher zu bilden, als mit den weitaus detaillierteren etwa der SysML. Etwas mehr formale Präzision verstärkt die semantische Aussagekraft; eine ausführlichere Analyse wird in [Dun13] vorgenommen.

Fundamentale Modellelemente

Das ‚Fundamental Modeling Concept‘ (FMC) reduziert die Modellelement-Typen auf [Wen01, Knö05]:

- *Aktor* für aktive Elemente wie Rolle/Komponente/Funktion,
- *Zustand* für passive Elemente wie Information/Form und
- ◆ *Ereignis*.

Diagramme („Pläne“) dreier Typen werden definiert, nämlich *Block-Diagramm* für den Systemaufbau, *Petri-Netz* für das Verhalten und *Entity/Relationship-Diagramm* für

Datenstrukturen und Wertebereiche. Als *bipartite Graphen* haben FMC Pläne vorteilhafte Qualitäten, u.a. lassen sie sich sauber hierarchisch verfeinern [Wen01]. Modellelemente können (und sollen) in Abhängigkeit ihres Typs auf verschiedenen Diagrammtypen erscheinen (Fig. 5).

Neben den fundamentalen Modellelement-Typen des FMC werden für ein übergreifendes Systemmodell auch

- * *Merkmal* und
- ⚡ *Anforderung*

benötigt; damit enthält das vorgeschlagene Integrationsmodell insgesamt fünf Elementtypen.

Akzeptierte Diagrammtypen

Trotz überzeugender Eigenschaften und einer treuen Anwender-Gemeinschaft hat das FMC bisher keine große Verbreitung erfahren. Hinderlich bei der Verbreitung ist möglicherweise, dass in FMC neue, von gängigen Standards abweichende Graphen definiert werden.

Daher wird hier ein anderer Weg eingeschlagen. Die logische Prägnanz der fundamentalen Elementtypen Aktor, Zustand und Ereignis wird genutzt und für Modellelemente und Diagramme verbreiteter Methoden, beispielsweise BPMN und SysML verwendet. Damit ist eine Abstraktion der jeweiligen Notation entsprechend ihrer fundamentalen Eigenschaften verbunden. Die Notation bleibt für Autoren und Leser unverändert, während das Integrationsmodell sowohl formal wie inhaltlich prägnanter wird.

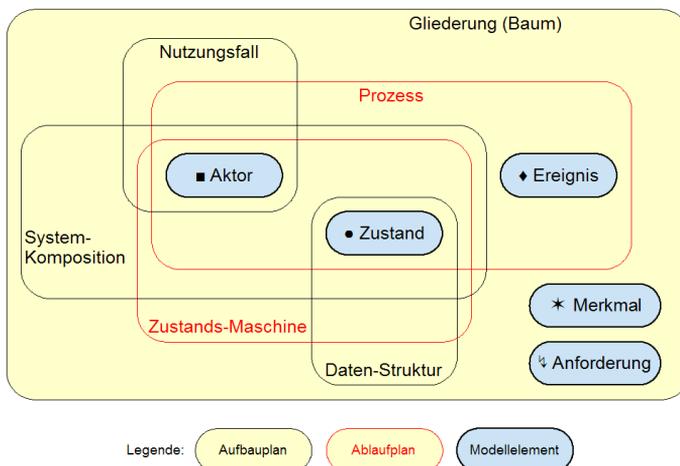


Fig. 5: Verwendung der Modellelement-Typen in Diagrammen/Plänen

Die Notation jedes Diagrammtyps bestimmt die verwendbaren Elementtypen (Fig. 5). Beispielsweise besteht eine typische Systemkomposition aus Funktionen und Datenspeichern, die den fundamentalen Typen Akteur und Zustand entsprechen. Ein Prozess wird mit Elementen der Typen Akteur, Ereignis und Zustand dargestellt. Eine Gliederung ist eine hierarchische Liste, die Elemente jeglichen Typs enthalten kann: Mechanische Systemkomponenten ebenso wie Software-Funktionen, Anforderungen oder Diagramme.

Modelle mit Bedeutung

Die Bedeutung des Systemmodells wird von den *Elementen* mit textuellen Beschreibungen und Attributen sowie von *logischen Beziehungen* zwischen Elementen getragen.

OEM-Partner	
OEM-Partner bieten im Rahmen der hausNETZ -Allianz eigene Geräte für das hausNETZ an. Der Eigentümer der Marke hausNETZ unterstützt die Geräteentwicklung der OEM-Partner und führt die Zertifizierung durch.	
Model Element Type	Actor
Reviewer Status	partly-agreed
Reviewer Comment	Die jeweiligen Verantwortungen der Vertragspartner sind noch nicht ausreichend beschrieben.
Author Status	to-evaluate
Author Comment	
Modified at	2015-07-12 18:57h
Modified by	od

Fig. 6: Datensatz eines Modellelements vom Typ ‚Akteur‘ mit Titel, Beschreibung und Attributen.

Jedes einzelne Modellelement besitzt einen Datensatz mit Beschreibung aus formatierten Texten und ggf. eingebetteten Illustrationen sowie einer Reihe von projektabhängigen Attributen (Fig. 6). Der Datensatz unterliegt der Volltextsuche, zusätzlich können Attribute mit Auswahlwerten (‚Enumeration‘) für eine Filterung herangezogen werden.

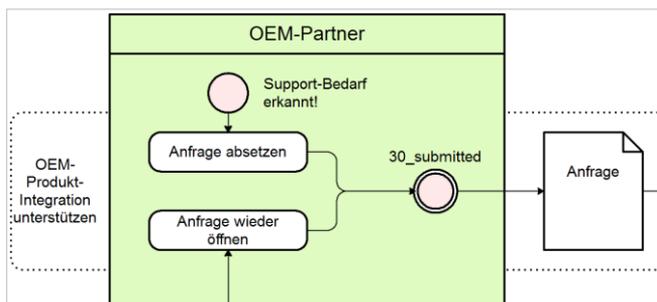


Fig. 7: Ausschnitt eines Modelldiagramms mit Prozess-Schritten und Ereignissen, hier gemäß ‚Business Process Model and Notation‘ (BPMN)

Diagramme veranschaulichen Aufbau, Abläufe oder Systemverhalten. Es werden die von den Beteiligten aus fachlichen Gründen ausgewählten Modellierungsmethoden mit ihrer jeweiligen Notation verwendet. Fig. 7 zeigt einen kleinen Ausschnitt eines Prozess-Modells. Die logischen Zusammenhänge, etwa die Abfolge der Prozess-Schritte oder die Zuordnung von Prozess-Schritten zu Rollen, erschließen sich nur dem Betrachter.

Aus einem Diagramm lassen sich zahlreiche logische Zusammenhänge automatisch bestimmen und als Relationen zwischen den betroffenen Modellelementen ablegen; so werden sie maschinell auswertbar. Die Diagrammelemente („Knoten“) werden in fundamentale Modell-Elemente transformiert. Im Beispiel entsprechen alle Rollen, Hauptprozesse und Prozess-Schritte dem fundamentalen Element-Typ ‚Aktor‘, während alle Dokumente und Nachrichten dem fundamentalen Element-Typ ‚Zustand‘ entsprechen. Aus den Verbindungen der Diagrammelemente („Kanten“) ergeben sich die logischen Beziehungen. Fig. 8 zeigt den entsprechenden Ausschnitt des Integrationsmodells.

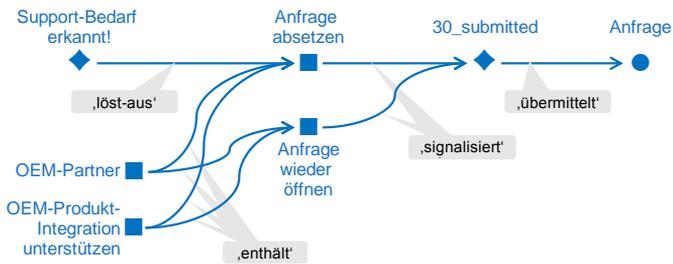


Fig. 8: Ausschnitt des Integrations-Modells zu Fig. 7 mit Elementen und automatisch abgeleiteten Beziehungen.

An dieser Stelle sei erwähnt, dass oft in einem Systemaufbau-Diagramm dargestellt wird, welche Funktionen eine Systemkomponente realisiert, so dass letztlich aus der Gesamtheit der abgeleiteten Relationen bestimmt werden kann, welche Rollen für welche Prozess-Schritte verantwortlich sind und welche Funktionen in welchen Systemkomponenten hierfür zur Verfügung stehen (sollen) [Dun13].

Neben diesen automatisch abgeleiteten Relationen können andere nur manuell erfasst werden. Beispielsweise ist die Zuordnung von Anforderungen zu Systemkomponenten das Ergebnis einer menschlichen Analyse.

Wie besprochen werden alle Elemente und Relationen gleich welcher Herkunft im Integrationsmodell zusammen gefasst. Widersprüche oder Lücken können erkannt und ihre Ursachen analysiert werden. Das Potential einer solchen systematischen und teil-automatischen Auswertung lässt sich an der Elektronik-Entwicklung erkennen, bei der graphische Schaltungsentwürfe formal in die Sprache ‚Very High Speed Integrated Circuit Hardware Description Language‘ (VHDL) transformiert und auf dieser Basis einer Prüfung anhand vielfältiger Entwurfsregeln unterzogen werden, bevor weitere Entwicklungsschritte eingeleitet werden. Trotz höchster Komplexität von Elektronik-Bauteilen und -Komponenten bieten die Hersteller seit Jahrzehnten eine sehr hohe Qualität bei erstaunlicher Innovationsgeschwindigkeit.

5. Ergebnisse der Praxis und Ausblick

Es gibt eine Reihe von Projekten, in denen integrierte Systemmodelle zur Produktkonzeption mit Erfolg eingesetzt wurden [Dun14]. Hierbei wurde ein Autorenwerkzeug auf der Basis des Fundamental Modelling Concepts (FMC) eingesetzt. Zur Beschreibung von Systemaufbau, Bediensequenzen und Modulverhalten wurden die Diagrammtypen FMC Blockdiagramm, BPMN Prozessmodell und Zustandsmaschine heran gezogen. Das Werkzeug unterhält einen Modellspeicher („Repository“), unterstützt aktiv die Wiederverwendung bzw. Konsolidierung von Modellelementen, interpretiert die Diagramme und hinterlegt automatisch entsprechende logische Beziehungen. Die Anforderungen werden in einer hierarchischen Liste erfasst und manuell den dargestellten Modellelementen zugeordnet. Das gesamte Systemmodell samt Anforderungen und Beziehungen wird im Requirements Interchange Format (ReqIF) exportiert und ist daher in vielen Systemen des Application/Product Lifecycle Management (ALM/PLM) nutzbar. Diese praktischen Beispiele erlauben den vorgestellten Weg zur Modellintegration nachzuvollziehen und zu bewerten.

Eine Integration von Modellen aus unterschiedlichen Quellen und Werkzeugen steht aus, ebenso die Abstraktion von Modellen der verbreiteten Notation SysML und deren Abbildung auf die fundamentalen Elementtypen des Integrationsmodells. Der Arbeitskreis PLM4MBSE der Gesellschaft für Systems Engineering (GfSE) hat sich die Aufgabe gestellt ein herstellernerutrales Format zur Repräsentation und zum Austausch von semantisch integrierten Systemmodellen zu erarbeiten und an Beispielen zu erproben [Kau15].

6. Zusammenfassung

Vier Darstellungs-Ebenen von Information im Rahmen der Konzeption mechatronischer Systeme wurden erläutert. Ein gangbarer Weg zur Integration von Modellen unterschiedlicher Quellen wurde zur Diskussion gestellt. Dabei wurden folgende vier Prinzipien der semantischen Integration angewandt:

1. *Trennung von Visualisierung und Modell*: Gliederungen, Listen und Diagramme sind Sichten eines gemeinsamen logischen Modellkerns.
2. *Abstraktion der Modellelemente*: Der Modellkern besteht nur aus Elementen der fundamentalen Typen Akteur, Zustand und Ereignis; hinzu kommen Merkmal und Anforderung. Spezifische Modelle werden gemäß der fundamentalen Eigenart ihrer Elemente in das übergreifende Informationsmodell des Modellkerns, den gemeinsamen Kontext, transformiert.
3. *Konsolidierung*: Identische Modellelemente auf Diagrammen gleichen oder unterschiedlichen Typs weisen auf ein einziges Element im Modellkern. In verteilten Modellen erhalten identische Modellelemente eine „same-as“ Relation.
4. *Vernetzung*: Logische Beziehungen zwischen Modellelementen werden durch explizite Relationen hinterlegt, teilweise automatisch und teilweise manuell.

Das Ergebnis ist ein umfassendes Systemmodell mit expliziten logischen Beziehungen, die maschinell auswertbar sind. Damit ist nicht nur eine effektivere Suche und Navigation in Information verschiedener Quellen möglich, sondern auch eine teilautomatische Prüfung.

Im Endeffekt äußert sich eine bessere Qualität der Systemkonzepte in effektiverer Zusammenarbeit verschiedener Teilorganisationen, sowie im Zeitgewinn durch früheres Auffinden fehlender und widersprüchlicher Angaben – mit allen damit verbundenen finanziellen Vorteilen.

Literatur

- [Wen01] Wendt, S.: Ein grundlegender Begriffsrahmen für das Wissensmanagement im Software-Engineering. In Proceedings „Knowtech“ Dresden 2001. http://www.community-of-knowledge.de/fileadmin/user_upload/attachments/f25.pdf.
- [Knö05] Knöpfel, A.; Gröne, B.; Tabeling, P.: Fundamental Modelling Concepts – Effective Communication of IT Systems. ISBN-13: 978-0-470-02710-3. John Wiley & Sons, Chichester, 2005.
- [Poh11] Pohl, K.; Rupp, Ch.: Basiswissen Requirements Engineering. ISBN 978-3-89864-771-7. dpunkt.verlag, Heidelberg, 2011.
- [Omg11] Object Management Group: Requirements Interchange Format (ReqIF). <http://www.omg.org/spec/ReqIF/>
- [Sml12] Object Management Group: OMG Systems Modeling Language (OMG SysML™), Version 1.3, <http://www.omg.org/spec/SysML/1.3/>, June 2012.
- [Dun13] Dungern, O.v.: Semantic Model Integration for System Specification – Creating a Common Context for Different Model Types, <http://enso-managers.com/files/ReqIF/resources/documents-en/Semantic-Model-Integration.pdf>.
- [Dun14] Dungern, O.v.: Übergreifende Konzeption von Geräten für die Gebäudeautomation – Methodik und Management. TdSE Tag des Systems Engineering der GfSE, Bremen, November 2014. [http://enso-managers.de/files/resources/enso-m/documents-de/TdSE-2014_Dungern_Uebergreifende-Systemkonzeption_\(Text\).pdf](http://enso-managers.de/files/resources/enso-m/documents-de/TdSE-2014_Dungern_Uebergreifende-Systemkonzeption_(Text).pdf).
- [Kau15] Kaufmann, U., Pfenning, M.: 10 Theses about MBSE and PLM, http://gfse.de/Dokumente_Mitglieder/ag_ergebnisse/PLM4MBSE/PLM4MBSE_Position_paper_V_1_0.pdf
- [Wik15] Wikipedia, <http://de.wikipedia.org>, abgerufen am 3.7.2015.