

Übergreifende Konzeption von Geräten für die Gebäudeautomation – Methodik und Management

Oskar von Dungern

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

Zusammenfassung: Ein innovativer Hersteller der Gebäude-Automation will sein erfolgreiches Produkt-Portfolio durch eine Familie drahtlos vernetzter Sensoren, Aktoren und Steuerungen erweitern. Dazu sind 120 systemfähige Geräte parallel zu entwickeln. Zugleich sind minimale Entwicklungskosten und –zeit gefordert, um die Vermarktungsziele in einem schnell wachsenden Markt zu erreichen.

Erfahrene Maschinenbauer, Elektrotechniker und Software-Entwickler, zuvor in kleinen Teams jeweils verantwortlich für ‚ihr‘ Gerät, entwerfen nun gemeinsam eine übergreifende Systemarchitektur und müssen ihre eigenen Geräte-Konzepte unterordnen. 1300 Anforderungen sind zu bewerten, im Aufwand zu schätzen und zu realisieren. Ein semantisch integriertes Systemmodell wird aufgebaut und auf Vollständigkeit, Konsistenz und Machbarkeit geprüft. Die Lastenhefte für alle 120 Geräte werden aus dem Gesamtmodell erzeugt. Entwickler und Tester nutzen eine gemeinsame Referenz vernetzter Anforderungen und Konzept-Diagramme.

Methode, Management und Aspekte der Beteiligten werden beleuchtet; zudem Ergebnis und Nutzen nach 15 Monaten des organisatorischen Wandels.

1. Aufgabenstellung

Systemspezifikationen spielen eine entscheidende Rolle als Ausgangspunkt in der Produktentwicklung. Sie sollen für eine effektive Kommunikation zwischen allen Beteiligten sorgen und als Referenz für Entscheidungen oder Entwicklungsergebnisse dienen. Dafür müssen sie verständlich sein und eine hohe Qualität aufweisen.

Aus methodischer Sicht sind anspruchsvolle Systeme mit Texten allein nicht hinreichend genau zu beschreiben. In der Praxis werden daher zunehmend strukturierte Anforderungslisten und Modelle verschiedener Art heran gezogen; dies wird in den folgenden Kapiteln näher untersucht.

Aus inhaltlicher Sicht stammen eingehende Informationen oft aus verschiedenen Teilorganisationen und beleuchten etwa optimierte Bediensequenzen, Anforderungen aus der Produktstrategie, Gesetze und Verbraucherschutz. Um Informationen und Modelle verschiedener Quellen zusammenführen zu können, ist ein gemeinsamer Kontext zu schaffen. Dabei werden folgende Ziele verfolgt:

- 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,
- Fehler oder Verletzung von Entwurfsregeln erkennen,
- Die interdisziplinäre Systemkonzeption unterstützen.

Der im weiteren Verlauf beschriebene Ansatz ist bewährt: Die Integration verschiedener Modellsichten hat ihren Nutzen in vielen Industrieprojekten gezeigt. Die Integration von Modellen aus unterschiedlichen Werkzeugen ist ein natürlicher nächster Schritt.

2. Stand der Technik

Für die Systemspezifikation haben strukturierte Anforderungslisten klare Vorteile gegenüber Texten. Allerdings ist es in umfangreichen Projekten immer noch sehr schwierig zu beurteilen, ob eine Liste vollständig genug und konsistent ist und ob sie mit vernünftigen Kosten realisiert werden kann. Dies ist eine grundsätzliche Schwäche heute erhältlicher Anforderungsmanagement-Werkzeuge.

Spezifikationen unter Verwendung von Systemmodellen bieten mehr, indem komplexe Systeme modularisiert, ihr dynamisches Verhalten beschrieben und ihre Fachdaten strukturiert werden können. Speziell UML und SysML kommen oft zum Einsatz, doch findet man in der Praxis leider häufiger ‚schlechte‘ Systemmodelle als verständliche und nützliche. Viele Diagramme werden erzeugt, aber ihr inhaltlicher Zusammenhang ist gering; das Modell hat viele ‚offene Enden‘. Tim Weilkiens schreibt in einem Blog: “A model is built with a model language. Typically the users focus is on the diagram and notation – the so called concrete syntax. Much more important is the world behind. There is the abstract syntax and semantic of the model elements. Simply spoken the structure and the meaning. The abstract syntax is the enabler to do computer-based analysis, simulation, traceability and so on. That’s the difference between a model and a drawing.”

Demnach steht bei der Modellierung meist die graphische Repräsentation im Vordergrund, während der Semantik und den logischen Abhängigkeiten zwischen den Diagrammen wenig Aufmerksamkeit geschenkt wird. Je mehr Element-Typen und Auswahloptionen eine Methode anbietet, desto schwieriger wird es in einer gegebenen Situation eine eindeutige Auswahl zu treffen. 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].

Semantische Unschärfe überlässt es Werkzeug-Herstellern und Nutzern den Inhalten eine Bedeutung beizumessen: Der Wert einer Spezifikation sinkt, wenn die Interpretation bei Autoren und Lesern voneinander abweicht. Zudem müssen Zusammenhang und logische Konsistenz durch Geisteskraft und Disziplin erzeugt werden; hierbei geben viele Werkzeuge im Auslieferungszustand wenig Unterstützung.

3. Potential der semantischen Vernetzung

Ziel ist ein Gesamtmodell, aus dem sich konsistente Auszüge für unterschiedliche Verwendung erzeugen lassen (Bild 1), sei es als druckbares Dokument (PDF) oder als Web-Seite (HTML). Im später beschriebenen Hausautomatisierungs-System werden aus dem Systemmodell z.B. 120 Lastenhefte für die einzelnen Geräte generiert.

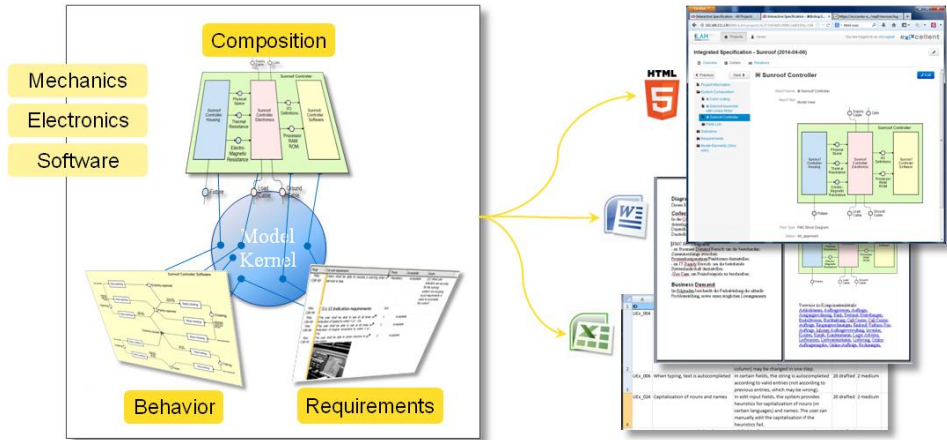


Bild 1: Logischer Modellkern mit Sichten und Ausleitung in verschiedenen Formaten

Im ersten Schritt wird aus Diagrammen und Listen unterschiedlicher Arten und Quellen ein gemeinsamer Modellkern gebildet. Dabei werden identische Modellelemente konsolidiert: Darstellungen auf verschiedenen Diagrammen weisen nun auf die gleichen Modellelemente (Bild 1). Namensänderungen werden sofort und überall sichtbar; Element-Beschreibungen gibt es nur einmal.

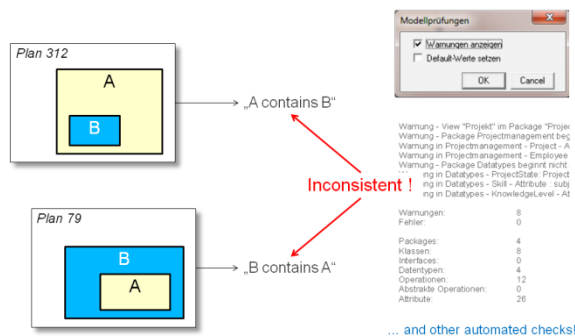


Bild 2: Automatische Modellprüfung bezüglich Enthaltenseins von Teilkomponenten

Sodann werden logische Beziehungen zwischen den Modellelementen hergestellt und auf Widersprüche untersucht. Beispielsweise kann automatisch ermittelt werden, wenn ein Plan eine Komponente ‚B‘ als Teil von ‚A‘ zeigt und ein anderer umgekehrt (Bild 2). Oder die Zuordnung von Anforderungen zu Systemmodulen hilft ihre Vollständigkeit

und Realisierbarkeit zu beurteilen. Zum Vergleich: Elektronik-Schaltungen werden bereits seit vielen Jahren formal beschrieben und automatisch auf Entwurfsfehler geprüft – das hat trotz hoher Entwicklungsgeschwindigkeit zu einer außerordentlichen Qualität der Produkte beigetragen.

Die Konsolidierung gleicher Modellelemente in verschiedenen Plänen und das Ableiten expliziter logischer Beziehungen wird hier als *semantische Vernetzung* bezeichnet. Damit sind viele praktische Vorteile verbunden:

- Das semantische Netz kann für die Navigation und die Prüfung heran gezogen werden. Z.B. sind die Anforderungen, die von einer bestimmten Systemkomponente zu erfüllen sind, sofort ersichtlich.
- Ein Modellkern hilft redundante Informationen zu vermeiden. Zugleich wird die konsequente Ausprägung von Gleichteilen erleichtert; im Beispiel wird das ‚Wireless Module‘ von einem Großteil der Geräte genutzt.
- Inkonsistenzen lassen sich aufdecken, teilweise sogar automatisch.

4. Übergreifende Systemkonzeption und 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 von SysML. Etwas mehr formale Präzision verstärkt die semantische Aussagekraft; eine ausführlichere Analyse wird in [Dun13] vorgenommen.

Das ‘Fundamental Modeling Concept’ (FMC) reduziert die Modellelement-Typen auf *aktive Elemente* für Akteure/Funktionen, *passive Elemente* für Zustände/Informationen sowie *Ereignisse* [Wen01, Knö05]. Diagramme dreier Typen („Pläne“ genannt) werden genutzt, nämlich Blockdiagramme für den Systemaufbau, Petri-Netze für das Verhalten und Entity/Relationship Diagramme für Datenstrukturen. Modellelemente können (und sollen) in Abhängigkeit ihres Typs auf verschiedenen Diagrammtypen erscheinen.

Modellelement	Plan					
	Gliederung	Prozess	Zustands- maschine	System- komposition	Organi- gramm	Entitäten/ Relationen
■ Akteur	X	X	X	X	X	
● Zustand	X	(X)	X	X		X
◆ Ereignis	X	X	X			
* Produktmerkmal	X					
⚡ Anforderung	X					

Tabelle 1: Verwendung der Modellelement-Typen in verschiedenen Plänen

Viele Praktiker ziehen traditionelle Prozessmodelle oder andere Darstellungsformen in Abhängigkeit ihres Fachgebiets vor, so dass angeraten ist die akzeptierten Diagramm-

typen zu nutzen. Deren Modellelemente werden beim Aufbau des Modellkerns in die genannten drei fundamentalen Typen überführt. Zusätzlich werden die Element-Typen *Produktmerkmal* und *Anforderung* verwendet. Tabelle 1 zeigt die resultierenden fünf Elementtypen und ihre Verwendung in verschiedenen Diagrammtypen.

Der Ansatz ist in mehrerer Hinsicht ‚übergreifend‘. Erstens setzt er dynamische Ablaufstrukturen, statische Aufbaustrukturen und Anforderungen in Beziehung. Zweitens stellt er Mechanik-, Elektronik- und Software-Komponenten in einen gemeinsamen Kontext. Drittens bietet er sich an für die Beschreibung von Systemfamilien mit Gleichteilen; dieser letzte Gesichtspunkt wird hier nicht weiter vertieft.

Zusammenfassend hat es sich nach unserer Erfahrung bewährt folgende vier Prinzipien der semantischen Integration anzuwenden:

1. *Trennung von Darstellung 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 Produktmerkmal 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.
4. *Vernetzung*: Logische Beziehungen zwischen Modellelementen werden durch explizite Relationen hinterlegt, teilweise automatisch und teilweise manuell.

5. Anwendung: Systemmodell eines Dimmers

Zur Illustration wird nachfolgend das Systemmodell eines Dimmers, Teil des Baukastens für Hausautomatisierungssysteme, vorgestellt. Es weist dynamische und statische Sichten auf, die auf verschiedene Weise semantisch miteinander verknüpft sind.

Der Dimmer hat zwei Bedientasten und wird durch kurzes oder längeres Drücken eines der beiden Taster betätigt. Die vier vorgesehenen *Bediensequenzen* werden hier in einem Prozessmodell nach BPMN dargestellt (Bild 3).

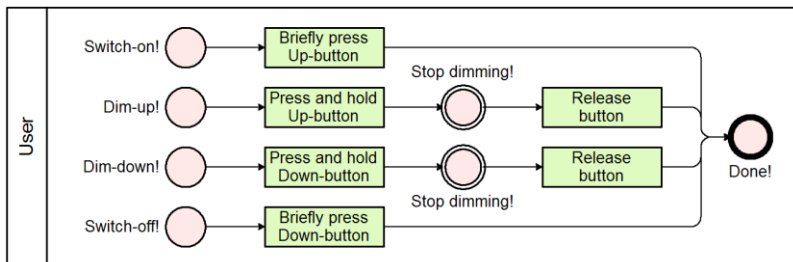


Bild 3: Die Bediensequenzen für Dimmer in allen Bauformen

Der *Systemaufbau* des Dimmers in der Unterputz-Bauform zeigt alle Module und ihre Schnittstellen (Bild 4). Hier wird ein FMC Blockdiagramm verwendet; für diesen Zweck kann auch ein SysML Komponentendiagramm gewählt werden.

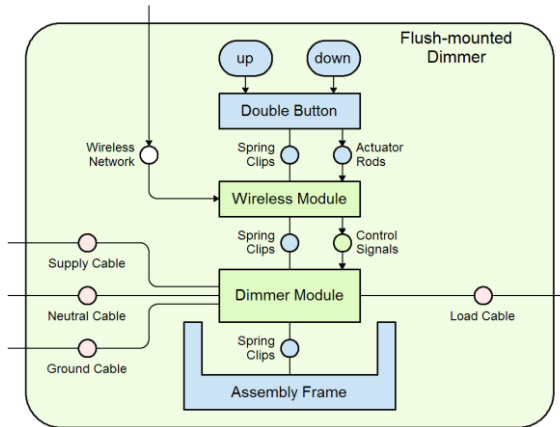


Bild 4: Systemaufbau des Unterputz-Dimmers

Die im ‚Wireless Module‘ untergebrachte ‚Button-Observer-Software‘ setzt die Tastendrucke in Steuerkommandos um; sie wird hier nicht näher beschrieben. Hingegen wird der Aufbau des Dimmer-Moduls weiter verfeinert (Bild 5); dabei werden die Komponenten der Mechanik für das Gehäuse, Elektronik für die Steuerungsplatine und Software für die Steuerung mit ihren Schnittstellen erkennbar. Beispielsweise ist der verfügbare Innenraum des Gehäuses zugleich der Bauraum für die Elektronik und wird daher als Schnittstelle dargestellt; die entsprechenden Daten und Zeichnungen sind über das Modellelement ‚Physical Space‘ erreichbar.

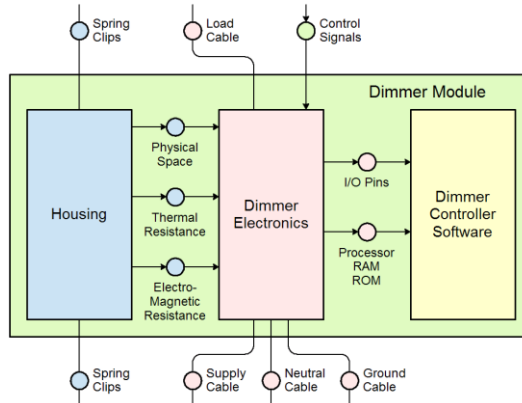


Bild 5: Komponenten und Schnittstellen des Dimmer-Moduls

Die Dimmer-Steuerung ist eine Software-Komponente, die auf die Steuerkommandos reagiert und die angeschlossene Leuchte heller oder dunkler werden lässt, bzw. ein- oder

ausschaltet. Zur Beschreibung ihres dynamischen *Verhaltens* dient hier ein Petri-Netz; eine spezielle Darstellung von Zuständen und ereignis-getriggerten Transitionen (Bild 6). Die Aussagekraft reicht aus, um das Verhalten bei allen Zuständen und für alle Sequenzen von Steuerkommandos zu simulieren, so dass beispielsweise die Blockade-freiheit bewiesen werden kann, bevor die Realisierung beginnt.

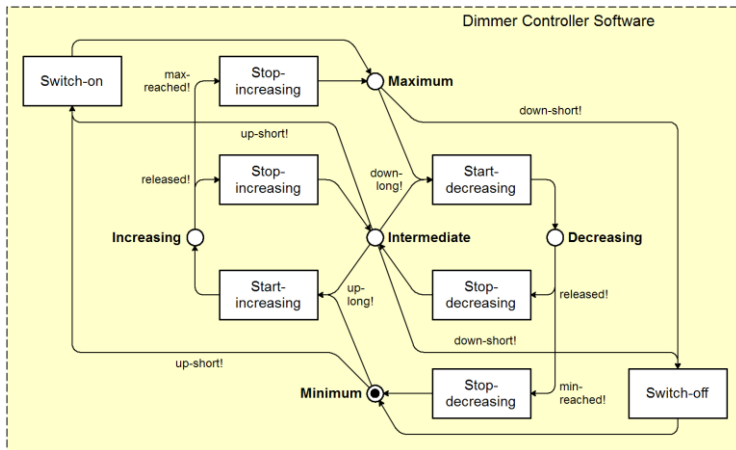


Bild 6: Verhalten der Dimmer Steuerung bei Steuerkommandos

Der Modellkern hinter den Modelldiagrammen wird teil-automatisch aufgebaut (Bild 7). Beispielsweise wird eine Relation ‚writes‘ angelegt, wenn ein Akteur eine gerichtete Verbindung mit einem Zustand hat. Hingegen müssen Relationen des Typs ‚satisfies‘ zwischen Komponenten und Anforderungen manuell angelegt werden.

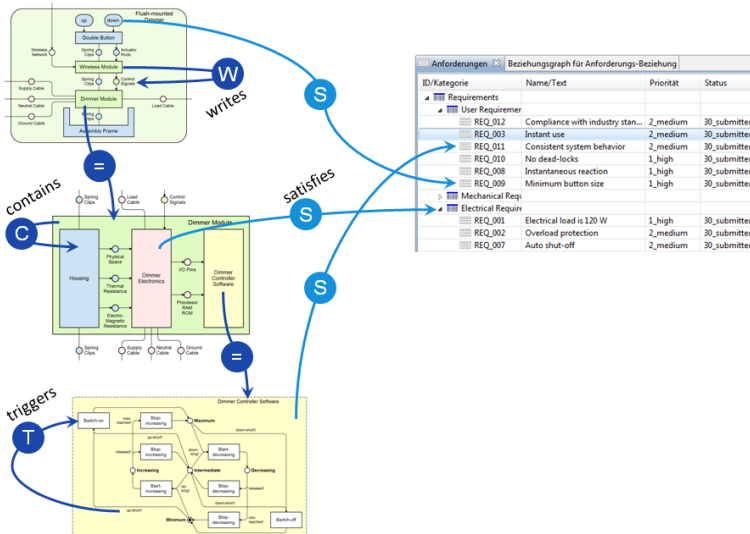


Bild 7: Einige logische Relationen zwischen Modellelementen

Darüber hinaus wird automatisch eine Beziehung ‚shows‘ zwischen einem Plan und jedem gezeigten Modellelement angelegt; so ist einfach zu ermitteln auf welchen Plänen ein bestimmtes Element vorkommt. In der Praxis relevante Relationen und ihre Aussagekraft werden in [Dun13] detaillierter beschrieben.

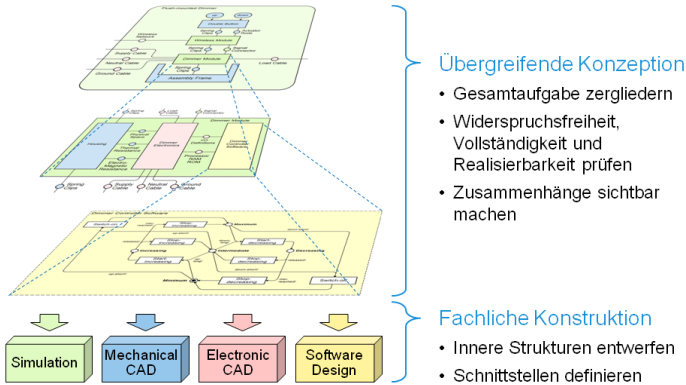


Bild 8: Komponenten übergreifend konzipieren und an die Konstruktion übergeben

Komponenten auf einem Diagramm werden bei Bedarf hierarchisch verfeinert. In elektronischen Medien lässt sich per Klick mittels eingebetteter Verweise von einem zum anderen Diagramm navigieren (Bild 8). Auf der niedrigsten Ebene erreicht man eine Elementbeschreibung mit projektspezifischen Attributen und den zuvor beschriebenen logischen Beziehungen. Sobald ‚reine‘ Komponenten identifiziert sind, kann die Konstruktionsaufgabe in das fachspezifische Entwicklungswerkzeug übernommen werden; ein Verweis mittels web-link bietet sich an.

6. Infrastruktur für den Entwicklungsprozess

Eine Spezifikations-Plattform übernimmt Anforderungen, Ablauf- und Aufbau-Modelle aus verschiedenen Quellen mittels standardisierter Datenformate (ReqIF o.a.). Sie gibt jedem Modellelement eine Web-Adresse, um eine Vernetzung mit anderen Daten oder Dokumenten sowie einen gezielten Zugriff mittels Web-Browser zu ermöglichen.

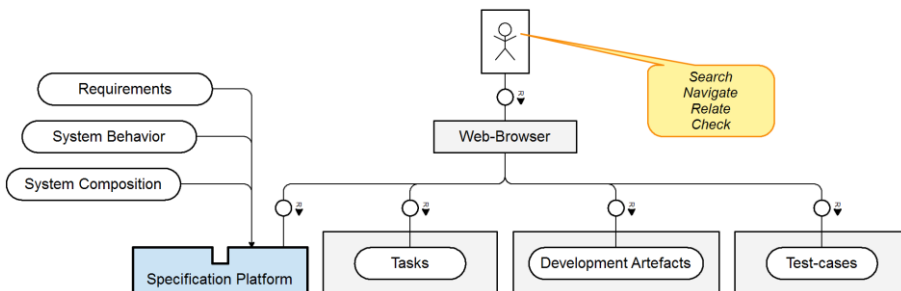


Bild 9: Werkzeugkette mit vernetzten Datenelementen

Während die Originale der Anforderungen, Ablauf- und Aufbau-Modelle weiter in ihren jeweiligen Autoren-Werkzeugen gepflegt werden, hält die Spezifikations-Plattform die essentielle Information und ihre semantischen Beziehungen in einem gemeinsamen Kontext; entfernt vergleichbar mit einem Index für Dokument-Management-Systeme.

So können Daten aus dem Task-Management, aus Entwicklungs-Umgebungen und aus dem Test-Management vernetzt werden – ohne das heute weit verbreitete Kopieren von Daten von Tool zu Tool. Die Spezifikations-Plattform wird zum Bezugspunkt, zum ‚Single-Point-of-Truth‘ in einer service-orientierten Architektur.

7. Management

Die Einführung des Systems Engineering bringt einen tiefgreifenden methodischen und organisatorischen Wandel mit sich. Als Wegbereiter wurden einige Mitarbeiter mit Freude an interdisziplinärem Denken und an Modellierung gefunden und ausgebildet.

Für die Systemkonzeption wurde ein interdisziplinäres Kernteam aus Produktmanager, Systemarchitekt, Mechanik-, Elektronik- und Software-Entwicklern sowie Wissensträgern zu Spezialgebieten wie Funktechnik gebildet. Mit etwas methodischer Unterstützung bei der konkreten Projektarbeit etablierte sich die neue Arbeitsweise nach wenigen Wochen; nach 5 Monaten waren fast alle der 1300 Anforderungen auf Systemkomponenten verteilt und ihre Realisierbarkeit bestätigt. Unter Kenntnis der Komponenten und ihrer jeweiligen Anforderungen konnten die Entwicklungsaufgaben abgeleitet und im Aufwand geschätzt werden. Etwa 5% Anforderungen wurden vom Produkt-Management überarbeitet, weil sie sich als Kostentreiber heraus gestellt hatten.

In der nachfolgenden Realisierung wurden die Entwickler der verschiedenen Sparten und das Testteam hinzu gezogen. Das online verfügbare Systemkonzept ist für Alle maßgeblich. Neue Anforderungen werden zunächst im System-Modell eingearbeitet und auf Verträglichkeit und Priorität geprüft. Nach der Freigabe werden die entsprechenden Aufgaben geplant und mit einem Verweis auf die relevanten Anforderungen versehen.

Für jede zu realisierende Komponente im Systemmodell wird ein entsprechendes Element in dem jeweiligen Entwicklungssystem für Mechanik, Elektronik bzw. Software angelegt und dort weiter ausgestaltet.

Einige erfahrene Ingenieure hatten durchaus Schwierigkeiten, ihre bislang weitgehend selbstbestimmte Funktion aufzugeben und eine Arbeitsweise anzunehmen, die von der Rücksicht auf eine Systemarchitektur und von gegenseitiger Abstimmung geprägt ist. In den meisten Fällen war ein koordiniertes Überzeugen seitens Beratung und Management schließlich erfolgreich; in wenigen war allerdings ein Personalaustausch mit anderen Projekten erforderlich.

Die Begleitung des Entwicklungs-Teams durch externe Berater wurde noch für etwa ein Jahr mit abklingender Intensität in Anspruch genommen.

8. Zusammenfassung

Die Konzeption eines umfangreichen Baukastens für die Hausautomatisierung erforderte neue Wege in methodischer und organisatorischer Hinsicht. Ein übergreifendes Systemmodell verbindet nun erstens die Mechanik-, Elektronik- und Software-Komponenten, zweitens dynamische Verhaltensmodelle, statische Aufbaumodelle und Anforderungen sowie drittens die einzelnen Geräte des Gesamtsystems. Die Modellelemente sind semantisch vernetzt, wobei viele Relationen automatisch aus der Anordnung der Modellelemente auf den Graphiken abgeleitet und andere manuell angelegt werden.

Die logische Vernetzung der Modellelemente erlaubt ein besseres Navigieren im Konzept, als es jeder Text oder jede Liste vermag, und ebenso ein weitaus effektiveres Prüfen der Zusammenhänge. Bestimmte logische Inkonsistenzen können zukünftig sogar automatisch entdeckt werden. Die Praxis zeigt, dass die Qualität der Systemspezifikation deutlich zugenommen hat.

Mit der vernetzenden Methodik ist ein organisatorischer Wandel verbunden. Interdisziplinäre Teams sorgen für eine optimale Systemgestaltung, deren Ergebnisse im gemeinsamen Systemmodell dokumentiert werden. Ein Mitarbeiter übernimmt die Gesamtverantwortung; aus einzelnen Teams wird eine Gesamt-Mannschaft geformt. Das gelingt bei klarer Zielsetzung und Unterstützung durch das Management, wobei eine vorübergehende Begleitung durch erfahrene Berater den Wandel beschleunigt.

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.fmc-modeling.org/publications>.
- [Knö05] Knöpfel, A.; Gröne, B.; Tabelaing, P.: Fundamental Modelling Concepts – Effective Communication of IT Systems. ISBN-13: 978-0-470-02710-3. John Wiley & Sons, Chichester, 2005.
- [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] Dungen, O.v.: Semantic Model Integration for System Specification – Creating a Common Context for Different Model Types, http://reqif.de/tl_files/ReqIF/resources/documents-en/Semantic-Model-Integration.pdf.