

Wissensbasiertes Assistenzsystem für modulares Engineering

Knowledge-Based Assistance System for Modular Engineering

Michael Obst*, Falk Doherr, Leon Urbas, Technische Universität Dresden

* Korrespondenzautor: michael.obst@tu-dresden.de

Zusammenfassung Die Zeitspanne von der Entwicklung bis zur Inbetriebnahme einer typischen Chemieanlage beträgt mit dem heutigen Stand der Technik ca. 5 bis 10 Jahre. Durch den Einsatz standardisierter Module sowie deren Auslegung mit Hilfe des fallbasierten Schließens sollen sich wiederholende Problemstellungen innerhalb der Planungsphase in deutlich kürzerer Zeit gelöst werden und somit zu einer Effizienzsteigerung und Kostenreduktion führen.

Summary The process from design up to operation of a typical chemical plant takes about 5 to 10 years with state-of-the-art technology. Recurring problems should be solved by using standardized modules to reduce the engineering time, become more efficient and to reduce the costs. The case-based reasoning follows the approach of comparing the actual case with similar cases from past experiences to achieve the solution faster.

Schlagwörter Modularisierung, fallbasiertes Schließen, Assistenzsystem, Engineering **Keywords** Modularization, case-based reasoning, assistance system, engineering

1 Einleitung

Im Lebenszyklus einer Anlage ist der Abschnitt der Planung durch umfangreiche Investitionen geprägt. Üblicherweise beläuft sich die Dauer von der Produktentwicklung bis zur Inbetriebnahme einer Anlage auf ca. fünf bis zehn Jahre [1]. Um eine qualitativ äquivalente Planung in einem kürzeren Zeitraum gewährleisten zu

können, ist die Entwicklung und der Einsatz neuer Methoden und Technologien erforderlich.

Als wichtigste Ansatzpunkte für die Verkürzung der Planungsprozesse in der Prozessindustrie wurden eine weitgehende Standardisierung, integrierte Informationsflüsse und Arbeitsprozesse sowie die Verwendung von Modulen identifiziert [1]. Idealerweise ersetzt ein

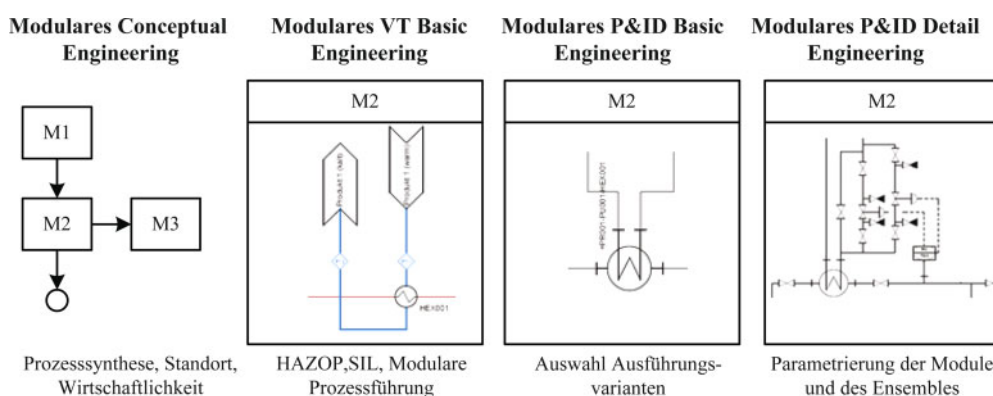


Bild 1 Modulares Engineering [2].

Auswahlprozess den Auslegungsprozess, der die Frage beantwortet, ob die geforderte Prozessfunktion mit einer bestehenden Modulbaugruppe wirtschaftlich umsetzbar ist (siehe Modul 2 (M2) in Bild 1) [2].

Der Einsatz standardisierter Module verspricht eine neue Problemstellung schneller bearbeiten zu können. Voraussetzung für ein gutes Ergebnis ist, dass i) das Modul eine gute Lösung für das Teilproblem darstellt, und ii) die ausgewählten Module in ihrer Gesamtheit eine Lösung für die Problemstellung darstellen. Ein Zeitgewinn entsteht, wenn das Suchen geeigneter Module und ihrer Kombination sowie die Anpassung an die aktuellen Anforderungen einen geringeren Aufwand mit sich bringen als der Entwurf einer Speziallösung. Erfahrungen mit Bibliotheks- und Typalkonzepten zeigen auf, dass hier rechnerbasierte Unterstützungssysteme sinnvoll sind [3]. Aus dieser Motivation heraus werden Modularisierungsansätze und das Fallbasierte Schließen im verfahrenstechnischen Kontext in Theorie und an praktischen Beispielen untersucht.

2 Modularisierung

Die Gestaltung des verfahrenstechnischen Anlagenbaus nach modularen Gesichtspunkten bringt den Vorteil einer schnelleren Fertigung der Anlage mit sich. Dies wird durch Wiederverwendung sowie Standardisierung erreicht [1]. Die Schwierigkeit bei der Definition von Modulen liegt in der Begründung der Grenzen zwischen den einzelnen Teilen. Es muss fundiert argumentiert werden, warum ein Planungsobjekt zu einem bestimmten Modul zugehörig ist. Dabei gilt es, die Kriterien der Modulerstellung zu beachten. Grundlegende Krite-

rien zur Modulbewertung sind innere Festigkeit und Abgeschlossenheit, geringe Kopplung, Geheimnisprinzip, Handhabbarkeit und Wiederverwendbarkeit [4]. Die Konstruktionslehre bietet mit dem Baukastensystem nach Pahl und Beitz [5] eine Definition für Module an. Ein technisches System setzt sich aus einer Kombination von Elementen (Modulen) eines Baukastens zusammen (Bild 2). Dabei besteht der Baukasten grundsätzlich aus Muss- und Kannbausteinen. Grundbausteine, die zur Erfüllung der Systemfunktion unerlässlich sind, gehören zu der Gruppe der Mussbausteine. Ein weiteres Element dieser Gruppe sind die Hilfsbausteine, die beispielsweise Verbindungs- und Anschlusselemente darstellen. Die Sonderbausteine aus der Gruppe der Kannbausteine beinhalten Zusatzfunktionen. Anpassbausteine werden benötigt, um das System an Randbedingungen wie beispielsweise unterschiedliche Druckniveaus von Versorgungsleitungen anzupassen. Sie gehören ebenfalls zu den Kannbausteinen. Trotz ausdifferenzierter Baukastensysteme können spezifische Aufgabenstellungen dazu führen, dass bestimmte Teile nur mit Hilfe einer Einzelanfertigung kostengünstig realisiert werden können. Systeme mit solchen Komponenten werden als Mischsystem bezeichnet, die spezifische Teillösung als Nichtbaustein.

Voraussetzung für eine modulare Planung in einem Baukastensystem sind die Abgeschlossenheit der Module, eine Beschränkung der Rückkopplung auf andere Module des Baukastensystems sowie eine Kompositionsmethodik, die es erlaubt, aus vorgeprüften Modulen ein zuverlässiges Gesamtsystem zusammenzustellen. Für die verfahrenstechnische und automatisierungstechnische Auslegung

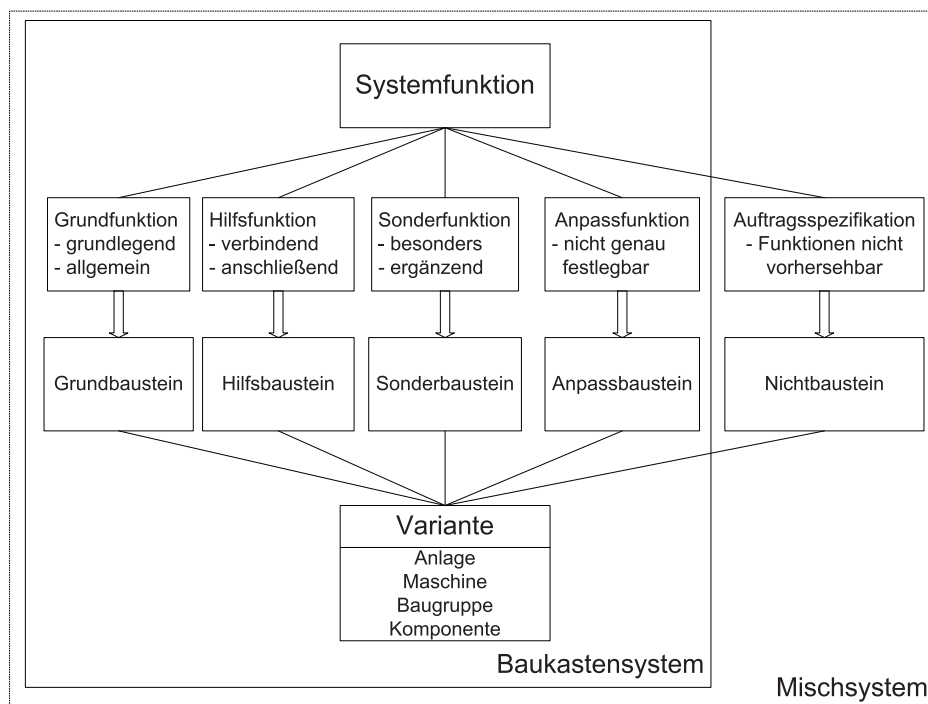


Bild 2 Baukastensystem der Konstruktionsmethodik [5].

bedeuten diese Anforderungen im Allgemeinen einen erhöhten Materialbedarf und einen größeren Umfang der Instrumentierung. Der erhöhte Materialbedarf resultiert aus einer für viele Fälle notwendigen Überdimensionierung. Der erhöhte Instrumentierungsbedarf ergibt sich aus der Forderung nach Minimierung der Wechselwirkungen zwischen den Modulen. Im Idealfall kann dieser Mehraufwand durch hohe Stückzahlen und Serienfertigung mindestens kompensiert werden.

Aufgrund der Besonderheiten der Prozessindustrie, wie stoffliche und energetische Kopplung über Modulgrenzen hinweg oder die hohe Abhängigkeit der Prozessführungsstrategie und der funktionalen Sicherheit von den chemischen und physikalischen Eigenschaften der Stoffgemische und -ströme, ist davon auszugehen, dass selbst für Produkt- und Prozessfamilien eine gewisse Anpassbarkeit der Mussmodule notwendig sein wird, um eine ökonomisch sinnvolle Stückzahl zu erreichen. Aufgrund des Faktors Zeit darf diese Anpassung jedoch nicht bedeuten, dass jedes Modul manuell durchgeplant werden muss – ein geeignetes Unterstützungssystem ist notwendig. Logikbasierte Lösungsansätze, beispielsweise auf der Basis von Prädikatenlogik, sind aufgrund der vielfältigen Ausnahmen, den Unterräumen mit eigenen Gesetzmäßigkeiten (Ex-Bereich, Material-Stoff-Wechselwirkungen, ...) und nichtlineare Eignungsfunktionen extrem aufwändig zu erstellen und zu warten. Modell-basierte Lösungsansätze, beispielsweise Entscheidungsbäume [6], müssen die Planungslogik der Prozessfamilie und Modulplattform abbilden, damit die zumeist binären Entscheidungsbäume handhabbar und selbst erklärend bleiben. Es bestehen enge Wechselwirkungen zwischen Konstruktionsmethode, Modulbeschreibung und modellbasiertem Verfahren, die einen hohen Aufwand bei der Integration neuer Erfahrungen nach sich ziehen.

3 Wissensbasierte Assistenzsysteme

Systeme zur Entscheidungsunterstützung basieren auf rationalen Prozessen oder systematische Prozeduren, die das Finden von ausgewogenen Entscheidungen ermöglichen. Diese können je nach Art der verwendeten Information und der eingesetzten Methoden als algorithmisch, wissensbasiert-induktiv und fallbasiert klassifiziert werden [7].

Algorithmische Ansätze definieren den Entwurfsprozess als Anwendung einer effektiven, domänenspezifischen Prozedur, die in aufeinanderfolgenden Schritten zu einer zufriedenstellenden Lösung führt. Als Beispiele seien hier Methoden zur Optimierung komplexer System durch Suchverfahren wie *breath-first*, *branch and bound* oder dynamische Programmierung [8;9] genannt.

Werden in einem Zustandsraum Wege von einer initialen Problembeschreibung zu einem Zielzustand definiert, so wird von einem wissensbasiert-induktiven Ansatz gesprochen. Dabei werden Übergänge von einem Zustand zu einem anderen durch die An-

wendung einer endlichen Menge von Operatoren erreicht. Diese Operatoren setzen sich aus funktionale Anforderungen und Entwurfsbeschränkungen (domänenspezifisches Wissen) und Metaregeln (domänenunabhängiges Wissen) zusammen. Diese Systeme werden häufig auch als „Expertensysteme“ oder „wissensbasierte Expertensysteme“ bezeichnet. Einen Überblick über den Einsatz dieser Systeme im Bereich Chemical Engineering geben Li und Kraslawski [10].

Fallbasierte Ansätze gehen davon aus, dass im Entwurfsprozess bereits gemachte Erfahrungen (Fälle) als Ausgangspunkte zur Lösung neuer Probleme herangezogen werden können [11]. Das Wissen besteht hier nicht aus generischen Regeln oder abstrakten Modellen, sondern aus einer Sammlung von Fällen (cases), in denen spezifische frühere Erfahrungen gespeichert sind. Ein neues Problem wird gelöst, indem ein ähnlicher Fall aus der Vergangenheit in der neuen Situation wiederverwendet wird. Ein weiterer wichtiger Unterschied zu regelbasierten Systemen besteht im Ansatz zum inkrementellen nachhaltigen Lernen, bei dem aus einer Problemlösung eine Erfahrung gewonnen wird, die für zukünftige Probleme zur Verfügung steht (als Positiv- und Negativbeispiel). Hier liegt ein weiterer Vorteil im Vergleich zu regelbasierten Systemen. Diese sind zwar in der Lage, Lösungen innerhalb ihres Beschreibungsraumes zu finden, neue Regeln müssen jedoch auf Widersprüche zu allen bereits vorhandenen Regeln überprüft werden, um so den Lösungsraum zu erweitern [11;12].

Aus den eben vorgestellten Ansätzen zur Entscheidungsunterstützung scheint die Methode des fallbasierten Schließens für die in Abschnitt 2 beschriebene Problemstellung am besten geeignet. Sie nutzt vorhandenes Wissen, erlaubt eine Strukturierung des Lösungsraumes und ist bedarfsgerecht erweiterbar.

Die zentrale Komponente des fallbasierten Schließens ist die Bestimmung der Ähnlichkeit zweier Fälle. Dabei ist es nur selten möglich, einzig aus dem Vergleich einzelner quantitativer Attribute auf die Ähnlichkeit von Fällen zu schließen. Fast immer müssen auch qualitative Aspekte mit berücksichtigt werden, die zum Beispiel durch Gewichtung und Abstraktion gewonnen werden können [13]. Zu diesem Zweck erfolgt eine formale Beschreibung der Fälle in Form von Attribut-Wert-Paaren. In komplexen Domänen hat sich die Einführung von Objekthierarchien und Graphen bewährt. Erstere erfordern eine geschickte Zusammenfassung der Attribut-Wert-Paare von Klasseninstanzen zu vergleichbaren Objektmerkmalen mittels geeigneter Amalgamierungsfunktionen [14], die graphbasierten Repräsentationen greifen auf heuristische Verfahren zur Bestimmung von Strukturähnlichkeiten zurück [15]. Die Fälle sind in Form einer Kombination aus Problem- und Lösungsbeschreibung in einer Fallbasis, dem Gedächtnis des Systems, abgelegt.

Zum praktischen Einsatz kommt das fallbasierte Schließen überwiegend in Anwendungssystemen für den

Kundendienst, in denen sie unter anderem zur Diagnose und Lösung von Kundenproblemen [16; 17], im Bereich der Technischen Diagnose [18] oder bei Prozessschritt- und Geräteauswahl [13] eingesetzt werden. Ein weiteres Anwendungsgebiet des fallbasierten Schließens ist das Design technischer Systeme. Hierzu muss ein solches System geeignet beschrieben werden. Eine explizite Beschreibung ermöglicht die klare Definition über den Aufbau und das Verhalten des Systems. Die implizite Spezifikation eines technischen Systems beschreibt Anforderungen und Funktionen, die ein System erfüllen muss [19]. Ist eine klare Trennung zwischen der Problembeschreibung und der korrespondierenden Lösung nicht möglich, können klassische CBR-Systeme [12] nicht zum Einsatz kommen [20].

Einfache Ähnlichkeitsmetriken wie zum Beispiel der euklidische Abstand reichen in vielen Anwendungen nicht aus, um die Ähnlichkeit zu bestimmen. Hier sind anwendungsbezogene Metriken notwendig [21].

4 Wissensbasiertes Assistenzsystem für modulares Engineering

Im Folgenden wird ein Ansatz vorgestellt, mit dessen Hilfe eine Beschleunigung des Engineerings eines verfahrenstechnischen Moduls erzielt werden soll. Weiterhin wird ein Vorschlag unterbreitet, wie dieser Ansatz in ein bestehendes CAE-System integriert werden kann.

4.1 Workflow

Die Definition der Anforderungen eines Moduls erfolgt in Form einer Funktionsdefinition. Ein Modul soll unter Berücksichtigung von Randbedingungen eine verfahrenstechnische Funktion wie beispielsweise das Dosieren erfüllen. Variante 1 ermöglicht das Dosieren mit Hilfe einer Pumpe mit konstanter Drehzahl und eines Stellventils. In Variante 2 sei die Drehzahl der Pumpe regelbar. Wird ein Modul zur Erfüllung einer bestimmten Funktion benötigt, können aufgrund einer Abfrage der Fallbasis alle Module herausgefiltert werden, die die geforderte Funktion erfüllen. Weitere Parameter, die die Eigenschaften eines Moduls darstellen, sind beispielsweise die erforderliche Druckstufe oder die Betriebstemperatur, ob ein Modul Cleaning in Place (CIP) oder Sterilization in Place (SIP) erfüllen kann, ob es ausbaubar ist oder einen Bypass besitzt. Dieser erste Abgleich der neuen Anforderungsbeschreibung mit der Fallbasis liefert alle Realisierungen eines Moduls, die die Spezifikation mit einer gewissen Ähnlichkeit erfüllen. In einem weiteren Schritt werden eventuell unvollständige Anforderungsbeschreibungen in der Fallbasis durch eine Analyse der vorhandenen CAE-Daten und der R&I Fließbilder aufgewertet.

4.2 Ähnlichkeit zweier Anforderungen

Die Bestimmung der Ähnlichkeit zweier Anforderungsbeschreibungen, die in diesem Fall der Problembeschreibung eines Falls entspricht, hat eine zentrale Bedeutung

zur Identifizierung alter Lösungen in der Fallbasis. Dabei bedeutet in diesem Fall die Beurteilung der Ähnlichkeit zweier Anforderungen eine Beurteilung darüber, wie nützlich eine Realisierung zur Erfüllung einer spezifischen Funktion ist. Eine Liste von Anforderungen, wie sie beispielhaft in Abschnitt 4.1 aufgeführt wurde, kann in einem Tupel

$$\mathbf{X} = (x_1, \dots, x_n)$$

zusammengefasst werden. Dabei beinhaltet jedes den entsprechenden Wert des Attributes. Diese Parameter werden mit Hilfe von spezifischen Ähnlichkeitsmetriken verglichen und bewertet. Die Funktion

$$\text{sim}_i(x_i, y_i)$$

vergleicht jeweils ein Attribut zweier Anforderungsbeschreibungen und bildet die Ähnlichkeit dieser zwei Eigenschaften auf einen Wertebereich von 0 bis 1, wobei 1 für identisch und 0 für nicht ähnlich steht. Beispiel solcher lokaler Ähnlichkeitsfunktionen können Tabellen oder mathematische Funktionen sein. Die gewichtete Summe der Ähnlichkeiten der einzelnen Anforderungen ergibt die Ähnlichkeit aller Anforderungen zweier Problembeschreibungen A und B.

$$\text{sim}(\mathbf{A}, \mathbf{B}) = \frac{\sum_i^n w_i \text{sim}_i(a_i, b_i)}{\sum_i^n w_i}$$

Bei dem Vergleich zweier Anforderungen an einem Modul ist es möglich, dass eine Eigenschaft eines Moduls in der Fallbasis nicht spezifiziert wurde. Am Beispiel der „Ausbaubarkeit“ eines Moduls bedeutet dies, dass ein Modul zwar technisch ausbaubar sein kann, dies jedoch keine Anforderung an die alte Realisierung war und somit die Anforderung nicht ausgefüllt wurde. Eine Besonderheit dieser Domäne ist, dass eine Reihe von Informationen über eine Funktion aufgrund der Darstellung auf dem R&I Fließbild und den dazu verfügbaren CAE-Daten gewonnen werden können, obwohl diese nicht direkt in der Anforderungsbeschreibung spezifiziert wurden. Die „Ausbaubarkeit“ eines Moduls ist zum Beispiel eine Eigenschaft, die für den Ingenieur aus dem R&I Fließbild und den CAE-Daten ablesbar ist (siehe Bild 3).

Das Modul auf der linken Seite ist aufgrund der beiden Ventile und der verbauten Muffen ausbaubar, das Ventil auf der rechten Seite jedoch nicht. Wird für ein neues zu lösendes Problem die Anforderung „Ausbaubarkeit“ gefordert und enthält die Anforderungsbeschreibung einer älteren Realisierung nicht die notwendige Eigenschaft, weil diese damals nicht gefordert war, so kann diese Realisierung die Anforderung eventuell erfüllen. Die Lösung



Bild 3 Beispiele für ein ausbaubares und ein nicht ausbaubares Modul.

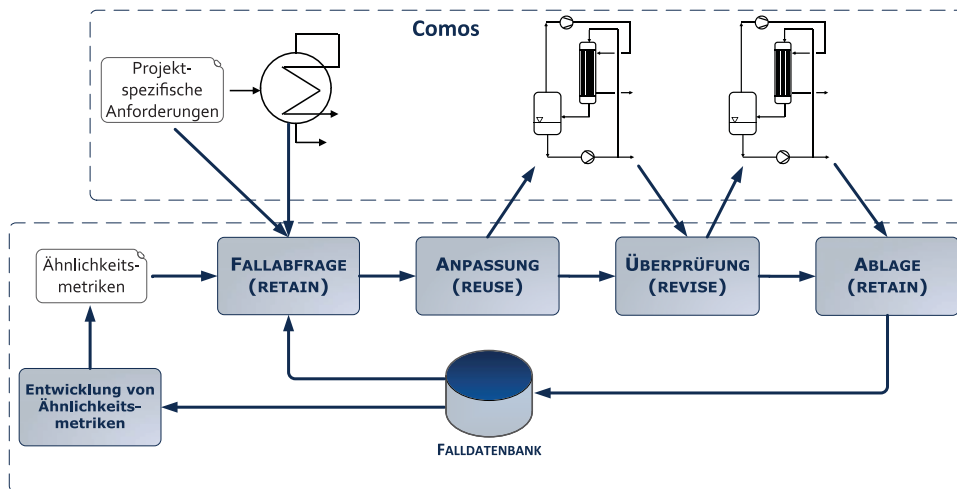


Bild 4 Systemkonzept Kopplung CAE- und CBR-System.

dieses Problems liegt in dem Vergleich der alten konkreten Lösung, mit einem hinterlegten Lösungsmuster für die die Eigenschaft „Ausbaubar“, wie es in Bild 3 auf der linken Seite dargestellt wurde. In dem konkreten Beispiel erfolgt der Vergleich über die Existenz von funktionalen Einheiten Absperrventil an den Materialflusskonnektoren der zu betrachtenden Einheit. Formal kann dieses Muster auch auf Eigenschaftsebene beschrieben werden – anstelle von Absperrventilen würde das Lösungsmuster Objekte oder Objektkombinationen in der Peripherie des Betrachtungsobjekts fordern, die die Eigenschaft „absperrbar“ erfüllen, sei es ein angesteuertes Schnellschlussventil oder eine Kombination aus Regelventil und Steckscheibe.

4.3 Anbindung an CAE-Systeme

Um ein wissensbasiertes Assistenzsystem während des Engineerings einer modularen Anlage einzusetzen, muss dieses in ein vorhandenes Planungswerkzeug eingebunden bzw. datenintegriert angebunden werden. Das an der Professur für Prozessleittechnik eingesetzten System Comos bietet hierfür eine Anbindung für externe Anwendung an vorhandene Planungsdaten. Mit diesem Werkzeug kann u. a. sowohl die Prozessauslegung, als auch das automatisierungstechnische Engineering auf einer Datenbasis durchgeführt werden. Mit Hilfe von nutzerdefinierten Abfragen und den integrierten XML-Connectoren ist es möglich, interne Planungsdaten von Comos in ein beliebiges XML-Format zu exportieren.

Die Planung erfolgt modular, das bedeutet, dass bereits in der Bibliothek des CBR-Systems die Module verfügbar auf Grund derer die Planung im P&ID Basic Engineering erfolgt. Im nächsten Schritt werden die Anforderungen des zu projektierten Moduls und gegebenenfalls projekt-spezifische Randbedingungen als Problembeschreibung an das CBR-System übergeben. Dies erfolgt mit Hilfe der XML-Connectoren. Für den modularen Im- und Export der Planungsdaten wurde ein auf dem CAEX Meta Model [22] basierendes Format definiert. Dieses enthält alle für die weitere Auslegung und Planung notwendigen

Informationen. Die Systemskizze in Bild 4 stellt die einzelnen Schritte des Fallbasierten Schließens und die jeweils notwendigen Interaktionen des Nutzers im CAE-System gegenüber.

5 Zusammenfassung und Ausblick

Das fallbasierte Schließen bietet die Möglichkeit, mit einer unvollständigen bzw. unscharfen Problembeschreibung zu einem Lösungsansatz zu gelangen. So kann bereits mit einem geringen Umfang an verfahrenstechnischen Informationen über ein Modul, wie zum Beispiel Drücke, Volumenströme oder Temperaturen, eine Aussage über den späteren Aufbau getroffen werden. Wenn ein bestimmter Parametersatz eines Moduls eine hohe Ähnlichkeit zu einem bereits in der Fallbasis vorhandenen aufweist, kann die dazugehörige (alte) Lösung mit einer hohen Wahrscheinlichkeit wiederverwendet und angepasst werden. Dies führt zu einem beschleunigten Engineering bei gleichbleibender Qualität und einer verbesserten Kostenschätzung, da Lösungen in der Fallbasis ebenfalls Kosteninformationen zur Verfügung stellen können. Dieser Ansatz wird an der TU Dresden untersucht und an praktischen Beispielen überprüft. Mit einem ersten Prototyp konnte bereits das Gesamtkonzept getestet werden, jedoch sind weitere Arbeiten an der Identifizierung geeigneter Referenzmodule und den Ähnlichkeitsmetriken notwendig.

Literatur

- [1] G. Schembecker, T. Bott: Die 50%-Idee Vom Produkt zur Produktionsanlage in der halben Zeit. PAAT-Jahrestreffen 2009, Weinheim, 2009.
- [2] L. Urbas, S. Bleuel, T. Jäger, S. Schmitz, L. Evertz: Automatisierung von Prozessmodulen Von Package-Unit-Integration zu modularen Anlagen. In: atp-Automatisierungstechnische Praxis, 54, Nr. 1–2, S. 44–53, 2012.
- [3] W. Marquardt, M. Nagl: Workflow and information centered support of design processes – the IMPROVE perspective. Computers and Chemical Engineering 29 S. 65–82, 2004.

- [4] B. K. Clark, Y. C. Baldwin: Design Rules, Volume 1 The power of Modularity. The MIT Press, 1999.
- [5] G. Pahl, W. Beitz: Konstruktionslehre, Springer, 1999.
- [6] G. Wozny; G. Schembecker: Informationstechnische Unterstützung der Anlagenplanung für die Angebots- und frühe Basic Engineering Phase durch ein modulares Planungskonzept, DE-CHMA, 2010.
- [7] Ch. Beierle, G. Kern-Isberner: Methoden wissensbasierter Systeme. Vieweg & Teubner, 2008.
- [8] B. Chandrasekaran: Design Problem Solving: A Task Analysis, AI Magazine 11(4), S. 59–71, 1990.
- [9] I. E. Grossmann, Z. Kravanja: Mixed-integer nonlinear programming approach for process systems engineering, Comp. & Chem. Eng. 19 (Suppl.) S. 189–204, 1995.
- [10] X. Li, A. Kraslawski: Conceptual process synthesis: past and current trends. Chemical Engineering and Processing, 43, S. 589–600, 2004.
- [11] J. L. Kolodner: Case-based reasoning. Kaufmann, San Mateo, Calif, 1993.
- [12] A. Aamodt; E. Plaza: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, 7, Nr. 1, S. 39–59, 1994.
- [13] Y. Avramenko; A. Kraslawski: Case Based Design: Applications in Process Engineering. Springer, 2008.
- [14] R. Bergmann; A. Stahl: Similarity measures for object-oriented case representations. In: B. Smyth; P. Cunningham (Hrsg.): Advances in Case-Based Reasoning. Springer, S. 25–36, 1998.
- [15] M. Dehmer; F. Emmert-Streib; J. Kilian: A similarity measure for graphs with low computational complexity. Applied Mathematics and Computation, 182, Nr. 1, S. 447–459, 2006.
- [16] W. Cheetham; K. Goebel: Appliance Call Center: A Successful Mixed-Initiative Case Study. Artificial Intelligence Magazine, 28, Nr. 2, S. 89–100, 2007.
- [17] R. Bergmann: Experience management: foundations, development methodology, and Internet based applications. Springer, 2002.
- [18] M. Devaney; W. Cheetham: Case-Based Reasoning for Gas Turbine Diagnostics. In: I. Russell; Z. Markov (Hrsg.): FLAIRS Conference. AAAI Press, S. 105–110, 2005.
- [19] B. Stein, M. Suermann, H. K. Büning, H.-W. Kelbassa, A. Reckmann, R. Tellmann, C. Thureau, H.-G. Wiegard: Fallbasiertes Schließen – Grundlagen und Anwendungen für Konstruktions- und Entwurfsaufgaben. University of Paderborn, Germany, Computer Science Institute, 2007.
- [20] K. Börner: Efficient Case-Based Structure Generation for Design Support, 2001.
- [21] A. Stahl: Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning, 2004.
- [22] IEC 62424: Representation of process control engineering request in P&ID diagrams and data exchange between P&ID tools and PCE-CAE tools, 2008.

Manuskripteingang: 16. Juli 2012



Dipl. Ing. Michael Obst ist wissenschaftlicher Mitarbeiter der Professur für Prozessleittechnik an der TU Dresden mit den Schwerpunkten: Unterstützungssysteme für modulares Anlagenengineering, Lifecycle Cost Engineering und Fallbasiertes Schließen.

Adresse: Technische Universität Dresden, Fakultät Elektrotechnik und Informationstechnik, Institut für Automatisierungstechnik, 01062 Dresden, E-Mail: michael.obst@tu-dresden.de



Dipl. Ing. Falk Doherr ist wiss. Mitarbeiter und Leiter der Arbeitsgruppe Funktions- und Informationsintegration an der Professur für Prozessleittechnik an der Technischen Universität Dresden und Projektgenieur bei der Linde Engineering Dresden GmbH. Sein wissenschaftliches Hauptarbeitsgebiet ist das integrierte prozessleittechnische Engineering mit Fokus feldnaher Kommunikationssysteme.

Adresse: Technische Universität Dresden, Fakultät Elektrotechnik und Informationstechnik, Institut für Automatisierungstechnik, 01062 Dresden, E-Mail: falk.doherr@tu-dresden.de



Prof. Dr.-Ing. Leon Urbas ist Inhaber der Professur für Prozessleittechnik an der Technischen Universität Dresden. Seine Hauptarbeitsgebiete umfassen Engineering verteilter sicherheitskritischer Systeme, insbesondere Funktionsintegration, modellgetriebenes Engineering, Modularisierung, Informationsmodelle der Prozessindustrie, Prozessinformations- und Managementsysteme und Middleware in der Automatisierungstechnik. Gebrauchstauglichkeit von multimodalen und mobilen Nahtstellen in Automatisierungssystemen, Analyse, Gestaltung und Bewertung von Alarmierungs- und Unterstützungssysteme sowie Methoden der Benutzermodellierung zur prospektiven Gestaltung von Mensch-Technik-Interaktion.

Adresse: Technische Universität Dresden, Fakultät Elektrotechnik und Informationstechnik, Institut für Automatisierungstechnik, 01062 Dresden, E-Mail: leon.urbas@tu-dresden.de