

Wissensmanagement für Ingenieure Die Pole-Position behaupten



Bild: BMW Motorsport

Wissen managen →

Tools & Begriffe

Communities

Anwendung

Barrieren

Portale

Baukästen

PDM

Die geflügelten Worte „Wenn Siemens wüsste, was Siemens weiß“ von Heinrich von Pierer* kann als Schlachtruf für eine neue, wissensbasierte Produktentwicklung gelten. Das Titelthema zeigt die unterschiedlichen Facetten von Wissensmanagement im Engineering auf.

* Der Vorsitzende des Vorstandes der Siemens AG, Heinrich von Pierer, anlässlich der Pressekonferenz „Siemens – The E-Driven Company“ am 10. Oktober 2000

DEFINITION

Tacheles reden

Der Begriff „Wissensmanagement“ findet heute in Unternehmen vergleichsweise inflationären Einzug. Wissensmanagement ist von den Führungsebenen bis hin zu den operativen Ebenen schlichtweg en vogue. Obgleich über die strategischen Ziele des Wissensmanagements – nämlich der Überführung einer unstrukturierten, zufallsbedingten Wissensnutzung in eine strukturierte und zielführende Nutzung – weitgehend Einvernehmen herrscht, divergieren die Vorstellungen über die Ansätze, Methoden und Hilfsmittel erheblich. Damit einhergehend ist die Erwartungshaltung bei der konkreten Implementierung von Wissensmanagement sehr unterschiedlich. Die erfolgreiche Implementierung bedingt aber ein gemeinsames Grundverständnis von Wissensmanagement.

Bei der Entwicklung eines gemeinsamen Verständnisses ist es hilfreich, Wissensmanagement als komplementären Zusammenschluss aus einem „organischen“ und einem „mechanistischen“ Wissensmanagement zu betrachten. Der Unterschied zwischen den beiden Komponenten macht sich anhand des jeweilig im Fokus stehenden Wissenszustands fest. Man unterscheidet pragmatisch zwischen „verborgenem“ und „explizitem“ Wissen. Ver- →

Lesen Sie bitte weiter auf Seite 24

Mit dem Aufkommen von Lean Management mussten wir Ingenieure eine teilweise erhebliche Beschränkung unserer Freiräume bei der Entwicklungsarbeit hinnehmen. Unser Tagesgeschäft war beherrscht von Verwaltungsarbeiten, weil das Management dem Glauben verfallen war, Outsourcing sei das Allheil-Mittel. Man überlies die Pflege und den Newerbwerb von Wissen anderen. Know-how in der Produktentwicklung wurde nicht als Kernkompetenz angesehen. Gegenwärtig jedoch ist der vielversprechende Trend zu beobachten, dass neue, marktreife Produkte und Dienstleistungen nur von Menschen entwickelt werden können, die über eine ganz bestimmte Expertise verfügen und sie diese Gewinn bringend umsetzen müssen. Erkennt ja, doch was verrät der Blick in die Praxis? Gehören nun die Arbeit am Fließband oder Alltagsroutine der Vergangenheit an? Jürgen Fuchs von CSC Ploenzke mahnt zur Vorsicht, sieht „die Gefahr des Erschlaffens von Wissen und Lösungsansätzen, wenn weiter Routine-Tätigkeiten in stark arbeitszerlegten Prozessen von den Beschäftigten gefordert und gefördert werden.“ Zuviel des Guten kann auch hier zum Nachteil gereichen.

Was verbirgt sich hinter „Wissensmanagement“? Die Antwort erscheint im Grunde einfach: All das, was eine lernende Organisation in ihrem Reifeprozess unterstützt. Wissen, wie und warum etwas zu realisieren ist. Der Teufel freilich liegt im Detail. Denn gerade die unorthodoxe Verknüpfung des Gewusstwie setzt den kreativen, innovativen Schaffensprozess in Gang. Es gilt, dem Ingenieur mehr Freiraum für seine Kreativität zu geben. Dies ist Aufgabe der Wissensbasierten Konstruktion (engl. Knowledge-Based Engineering, KBE). Darunter ist die Bereit-

stellung und Verarbeitung von Wissen mittels geeigneter IT-Applikationen zusammengefasst, die mit CAD-Systemen kooperieren und die für den Konstrukteur Teilaufgaben automatisch ausführen. Das an und für sich unwissende CAD-System wird durch programmiertes, so genanntes explizit formuliertes Wissen in seiner Leistungsfähigkeit wesentlich verbessert.

Das Warten auf den Kollegen, der gerade im Urlaub oder auf Dienstreise ist, aber auch die durch Globalisierung und Outsourcing zunehmende Arbeitsteilung stellen nur einige der Erschwerisse bei der Beschaffung von Wissen für die Produktentwicklung dar. Durch clevere Wissensbeschaffung lässt sich viel erreichen, gerade wenn man bedenkt, dass Konstrukteure einen erheblichen Teil ihrer Zeit mit der Suche nach Informationen verbringen (1). Die Welt ist für Konstrukteure kooperativer geworden. Damit hat auch der Verkehr auf den „Wissens-Highways“ zugenommen.

Regelmäßig sind im Konstruktionsprozess aufwändige Routine-Abläufe mit hohem Maß an Detailwissen durchzuführen. Beispiele hierzu sind Bauteil-Auswahl, Bemessung und Konstruktion einer perfekt ausgeführten Schraubenverbindung, die Festlegung von Maß-, Form- und Lagetoleranzen. Die konsequente Verwendung standardisierter Konstruktionen ist für viele Produkte ein Kosten bestimmender Faktor und deshalb möglichst zwingend zu beachten. Mit KBE-Applikationen lassen sich bekannte Routine-Tätigkeiten in einer Kom-

plexität automatisieren, wie es mit parametrisierten CAD-Modellen oder Verwendung von Konstruktionsfeatures nicht möglich ist. Mit KBE werden Konstruktionsprozesse programmiert und nicht nur skalierbare, parametrisierte Modelle im CAD-System angepasst.

Iterationsschleifen in der Entwicklung kosten in der Regel richtig Geld. Die steigende Anzahl der Produkte und Derivate, die der Markt heute fordert, führt zu einem weiteren Ansteigen der Iterationen. Die Kostenschraube dreht sich schneller. Hatte beispielsweise der Fahrzeug-Hersteller Audi im Jahr 1960 nur drei Modelle im Portfolio, sind es heute ein Vielfaches davon. Gab es in den 70er Jahren in jedem Haushalt den gleichen Teletyp, kann heute ein Handy nach sechs Monaten nur mehr schwierig verkauft werden. KBE kann wesentlich dazu beitragen, unnötige, das heißt vermeidbare Entwicklungsschleifen einzusparen. Die Produktentwicklung profitiert davon, dass ein virtuelles Produkt, repräsentiert durch CAD-Modelle und Metadaten in nachfolgenden Prozessschritten verwendet werden kann. CAD-Systeme verfügen grundsätzlich über eine Vielfalt von Möglichkeiten, ein Modell aufzubauen. Es sind jedoch Produkt spezifische Modellierungstechniken Voraussetzung dafür, ob das CAD-Modell einfach und schnell geändert oder in andere Programme übertragen werden kann. Mit speziellen KBE-Applikationen, so genannten Model Checkern, lässt sich die richtige Methodik in der Anwendung von CAD-Systemen trainieren.

Der Weg des Wissens ins IT-System

Herkömmliche Tools für Wissensmanagement und die Nutzung von Wissensmarktplätzen unterscheiden sich deutlich von wissensbasierter Konstruktion.

Bei Wissensmanagement-Systemen geht es primär um die Administration und schnelle, Personen unabhängige Bereitstellung von Wissen im Sinne einer Dokumentenverwaltung in den unternehmensweiten IT-Netzwerken. Diese Systeme helfen dem Anwender, relevante Informationen zu suchen und zu finden. Der Anwender selbst muss sie zu Wissen kombinieren, das heißt die Wissensverarbeitung findet im Kopf des

Anwenders statt. Nicht so bei der wissensbasierten Konstruktion.

Repräsentation von Wissen

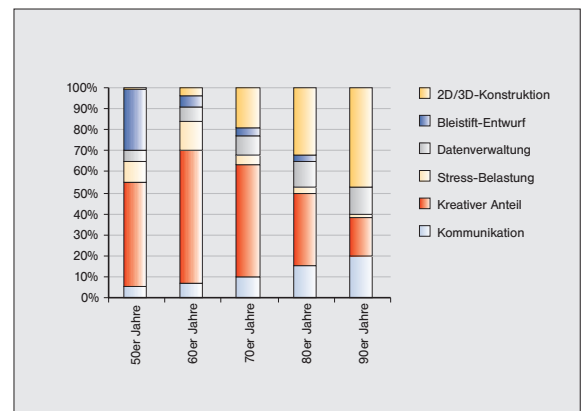
Grundlage zur Erstellung von wissensbasierten Systemen ist die Darstellung in einer Form, die die einfache Erfassung, Anwendung und Modifikation erlaubt. Eine Möglichkeit ist die Aufteilung in so genanntes deklaratives Wissen, das die Eigenschaften eines Objektes beschreibt und in prozedurales Wissen, welches die Aktionen beschreibt, die auf dieses Objekt angewendet werden können. Entsprechende Applikationen heißen auch Objekt orientierte Systeme.

EINFÜHRUNG

Wie der Erfolg garantiert ist

Folgende Maßnahmen haben sich bei der Einführung von Knowledge-based Engineering bewährt:

- Frühzeitige, permanente Einbindung des Managements. Wissensbasierte Konstruktion braucht die volle Unterstützung der Unternehmensleitung
- Einbindung aller Leistungsträger zur Spezifikation der wissensbasierten Applikationen
- Detaillierte Projekt-Planung und -Steuerung
- Gründung einer KBE-Initiative, die von allen Entscheidern und Betroffenen gleichermaßen verstanden, getragen und gelebt wird. Hier werden auch gemeinsam errungene Erfolge gefeiert

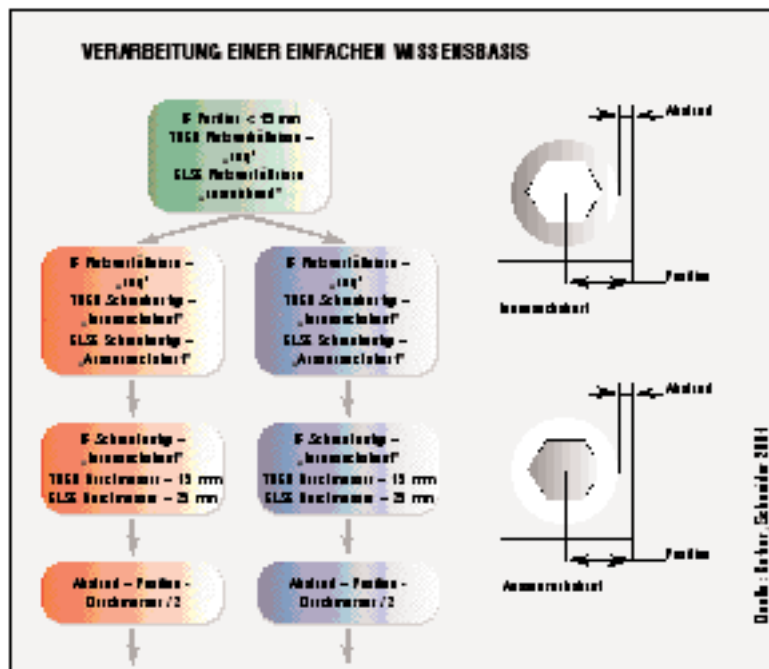


Der kreative Anteil am Ingenieursalltag (rot) hat in den letzten Jahrzehnten deutlich abgenommen. Siehe auch (4)

Wissensverarbeitung

Nicht nur komplexe technische Aufgabenstellungen, sondern auch scheinbar triviale Alltagsprobleme führen zu einer Vielzahl von Handlungsalternativen. Sie bedürfen der Abwägung. Dieser Entscheidungsprozess läuft bei den verantwortlichen Ingenieuren meist unbewusst ab. Bei der maschinellen Wissensverarbeitung jedoch ist sie von außerordentlicher Bedeutung. Sie manifestiert sich in der Auswahl und Anwendung der Suchstrategien zur Entscheidungsfindung. Die effektive Bearbeitung großer Suchräume durch geeignete Suchstrategien sind der Hauptgrund für den →

Die Verarbeitungsstrategie für ein Bauteil lässt sich im Rahmen der Grafentheorie als Zustandsgraf darstellen. Die Knoten im Zustandsgraf repräsentieren die Regeln, durch die Entscheidungen getroffen werden



Einsatz wissensbasierter Systeme als Alternative zu den höheren Programmiersprachen. Während in Fortran, C oder C++ jeder mögliche Entscheidungspfad explizit programmiert werden muss, übernehmen in wissensbasierten Anwendungen verschiedene Problemlösungsmechanismen diese Aufgabe und erleichtern so die Programmierung. Zusammen mit der Repräsentation von Wissen in leicht hand zu habender Form als Objektstrukturen ergibt sich eine effektive Vorgehensweise bei der Problemlösung.

Hierzu wird die Aufgabestellung nach den Vorschriften der Grafentheorie als Zustandsgraf formuliert. Eine mögliche Suchstrategie ist das so genannte Forward-Chaining, das von bekannten Daten ausgeht und alle möglichen Zwischen- und Endergebnisse ermittelt. Einen anderen Weg geht das Backward-Chaining, das ausgehend vom gewünschten Ziel den Zustandsgraf „rückwärts“ durchläuft und basierend auf Regeln und Zwischen-Ergebnissen die Daten ermittelt, die zum Erreichen des Ziels erforderlich sind.

Die bisher genannten Verfahren arbeiten in einer festen Reihenfolge den gesamten Zustandsgraf ab. Diese erschöpfende Suche ist aber nur bei überschaubaren Suchräumen geeignet, bei realistischen Aufgabenstellungen ergeben sich schnell Größenordnungen, die so nicht mehr zu Hand haben sind. Man

spricht deshalb auch von einer „kombinatorischen Explosion“.

Ziel ist es daher, die vorgestellten Verfahren so zu erweitern, dass die Suche im Zustandsgraf während der Abarbeitung dynamisch gelenkt wird. Heuristische Suchverfahren setzen dabei oft auf so genannte Kostenfunktionen. Das bedeutet, dass der aktuelle Zustand und die daraus resultierenden möglichen Folge-Zustände durch Zahlen bewertet werden. Der Pfad mit den jeweils größten Erfolgsaussichten wird dann weiterverfolgt. Weiterführende Lektüre hierzu findet sich beispielsweise unter (2,3).

Rule-Based Reasoning

Kernstück von Regel basierten Systemen ist die Wissensbasis. Sie enthält sowohl das applikationsspezifische Wissen in Form von If-Then-Regeln als auch Fall spezifische Informationen, also die Daten des aktuellen Programmlaufs. Regeln und Informationen werden durch die Inference Engine (Inference: Schlussfolgerung) je nach Aufgabenstellung durch eine der Suchstrategien zur Lösung des aktuellen Problems verarbeitet.

Model-Based Reasoning

Im Einsatz befindliche Regel basierte Systeme repräsentieren in vielen Fällen das Erfahrungswissen der Konstrukteure und nur selten die physikalischen Hintergründe der Entscheidungen, bei-

spielsweise die Theorie des Biegebalkens. Dies kann zu Situationen führen, in denen ein tieferes Problem-Verständnis zu besseren Lösungen geführt hätte, wenn der Ansatz des so genannten Model-Based Reasoning verfolgt worden wäre. Hierbei ist das Ziel, eine vollständige Simulation des Systemverhaltens zu erreichen, das heißt auch theoretisches Grundwissen in das wissensbasierte System einzubeziehen. In der Praxis sind solche Systeme beim Entwurf von logischen Schaltungen oder Netzwerken aus Operationsverstärkern im Einsatz.

Case-Based Reasoning

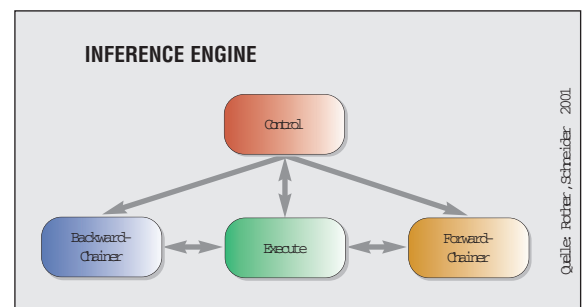
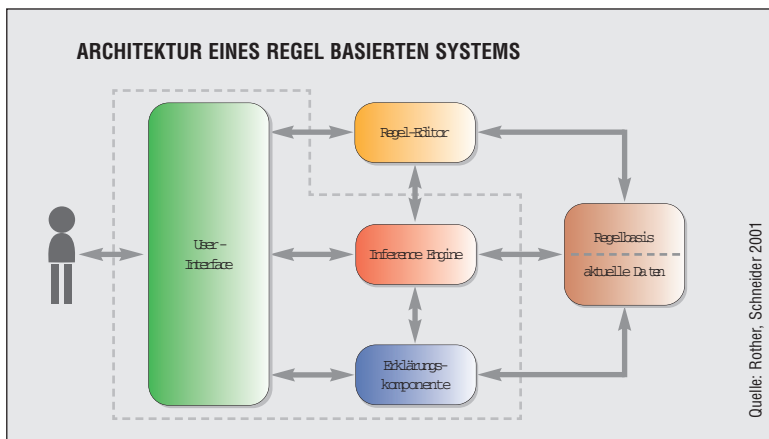
Eine weitere Strategie zur maschinellen Wissensverarbeitung ist die Schlussfolgerung aus Fallbeispielen („Case-Based Reasoning“). Diese Systeme speichern in einer Datenbank bereits aufgetretene Anforderungen und Probleme sowie deren Lösungen ab. Dadurch wird die Wissensakquise erleichtert, der Ingenieur muss nur zu einer Anzahl von existierenden Problemen die Lösung liefern, das System übernimmt dann die Verallgemeinerung der Regeln. Ein wichtiger Punkt ist dabei die sinnvolle Auswahl der Beispiele. Sie sollten allgemein anwendbar sein, also keine Spezialfälle darstellen.

Constraint-Based Reasoning

Versucht man alle möglichen Lösungen mit einem Zustandsgraf darzustellen, also auch Pfade mit offensichtlich falschen Lösungen zu berücksichtigen, spricht man von einer „Generate-and-test“-Vorgehensweise. Oft ist es jedoch vorteilhaft, das Wissen über bekannte Beschränkungen schon in die Vorgehensstrategie einzubeziehen. Systeme, die auf diesem Ansatz zur Einschränkung des Lösungsraums basieren, bezeichnet man als „Constraint-based“. Constraint-Based-Techniken finden Anwendung bei Aufgabenstellungen, die eine Entscheidungsfindung in Abhängigkeit von Zeiträumen wie bei Systemen zur Spracherkennung erfordern.

Fuzzy-Logik

Die bisher vorgestellten Systeme folgen generell dem gleichen Mechanismus: Ausgehend von korrekten Anfangsbedingungen werden neue, korrekte Er-



Typische Architektur eines Regel basierten Systems und die Inference Engine

gebnisse basierend auf eindeutig definierten Regeln erzeugt. Reale Problemlösungen lassen sich jedoch nicht immer auf widerspruchsfreie und in allen Fällen eindeutige Entscheidungsgrundlagen zurückführen. Trotzdem sind richtige Entscheidungen, die auf unvollständigen oder unsicheren Daten beruhen möglich – wie wir selbst jeden Tag aufs Neue beweisen. Ein Ansatz zum Umgang mit dieser Art von Daten sind Systeme, die auf Fuzzy Sets basieren. Die Werte „ja“ und „nein“ der klassischen (formalen) Logik werden durch reelle Zahlen zwischen 0,0 und 1,0 ersetzt, die in gewisser Weise die Wahrscheinlichkeit repräsentieren, dass der jeweilige Wert korrekt ist. Fuzzy Sets finden schon seit längerem eine weite Verbreitung als integrierte Systeme in Produkten der Konsumgüter-Industrie, etwa in Waschmaschinen oder Fotoapparaten.

Neuronale Netze

Ziel künstlicher neuronaler Netze ist nicht, Schlussfolgerungsmechanismen als Software in gängige Computersystemen zu implementieren, sondern die „Hardware“ des menschlichen Denkens, also das Gehirn, als Grundlage der Wissensverarbeitung abzubilden. Künstliche Zellen stellen das einfache Verarbeitungselement des neuronalen Netzes dar. Eine Zelle erhält gewichtete Eingangsinformationen über Eingangskanäle und erzeugt daraus innerhalb des „Zellkerns“ Ausgabe-Informationen, die über einen Ausgabekanal an benachbarte Zellen weitergeleitet werden. Charakteristisch ist die Tatsache, dass neuronale Netze für die gewünschten Aufgabenstellungen trainiert werden, eine explizite Programmierung ist unnötig.

Anbindung an CAD-Systeme

Es bestehen zwei grundsätzliche Möglichkeiten, wissensbasierte Systeme und CAD-Systeme zu verbinden. Hat das wissensbasierte System einen eigenen Modellierer, läuft der Prozess der Wissensverarbeitung und der Geometrie-Erstellung oder -modifikation unabhängig vom CAD-System ab. Es existieren dann Schnittstellen, über welche die Geometrie in das CAD-System übertragen, oder aus ihm herausgelesen wird. Als kommerzielles System fällt das System Icad von KTI in diese Kategorie. Der Vorteil liegt in der Unabhängigkeit vom jeweiligen CAD-System, der Nachteil in der generellen Problematik von Geometrie-Übertragungen wie unterschiedliche numerische Genauigkeit der Systeme, Differenzen bei der Repräsentation der Geometrie-Elemente et cetera. Der zweite Weg ist die direkte Integration in das CAD-System. Dabei nutzt das wissensbasierte System eine Programmier-Schnittstelle zum CAD-System. Die Geometrie wird somit durch das CAD-System erzeugt, wodurch die erwähnten Schnittstellen-Probleme entfallen. Ein weiterer Vorteil liegt in der Möglichkeit, die Benutzer-Oberfläche des CAD-Systems für das wissensbasierte System mit zu benutzen. Dies ist ein wichtiger Punkt für die Akzeptanz durch den Anwender. Allerdings ist das wissensbasierte System damit auf das gewählte CAD-System festgelegt, eine Erweiterung auf andere Systeme kann sehr aufwändig werden. Ein Beispiel zur industriellen Anwendung ist das Regelbasierte System Stonerule des Herstellers Planetcad im CAD-System Catia V4.

Es gilt, nicht nur ein anspruchsvolles technisches Problem in Form der Ent-

wicklung einer wissensbasierten Applikation zu lösen, sondern auch Veränderungsprozesse in Gang zu setzen, die sowohl den menschlichen Besitzständen und Ängsten („Wissen ist Macht“) als auch der Umsetzung veränderter Entwicklungsmethoden und der Entfaltung einer kooperativen Umgebung mit der Bereitschaft zum Geben und Nehmen Rechnung tragen. Das ist sicherlich keine leichte Aufgabe.

KLEMENS ROTHER, THOMAS SCHNEIDER

FEATURING

Die Autoren

Klemens Rother (links) und Thomas Schneider sind bei der CAD-FEM GmbH mit Sitz in Grafing bei München für den Bereich wissensbasierte Konstruktion tätig. Kontakt zu den Autoren über Tel. **+49-8092-7005-0** oder per E-Mail **krother@cadfem.de**



Literatur-Hinweise

- (1) Berliner Kreis, „Neue Wege zur Produktentwicklung“, Berlin 1997
- (2) Dengel, Andreas, „Künstliche Intelligenz: Allgemeine Prinzipien und Modelle“, Bibliographisches Institut & F.A. Brockhaus AG, Mannheim 1994
- (3) Luger, George F., Stubblefield, William A., „Artificial Intelligence: Structures and Strategies for Complex Problem Solving“, Addison Wesley Longman, USA 1998
- (4) CADplus 6/2000, Göller Verlag, Baden-Baden 2000