

Vergleich von tiefziehspezifischen Mess- und Simulationsdaten mit dem OpenForm-Postprozessor

Anwendungsbereich

Die Tiefziehsimulation ist ein wichtiges Hilfsmittel zur Auslegung von Tiefziehprozessen. Die Auslegung von Tiefziehprozessen auf Basis von Simulationsergebnissen ergibt selbstverständlich nur dann Sinn, wenn der simulierte Tiefziehprozess den realen Tiefziehprozess mit hinreichender Genauigkeit abbildet.

Bevor ein Tiefziehsimulationsprogramm in der industriellen Praxis zum Einsatz kommt, sollte eine Evaluierung bzw. ein Abgleich der Software auf Basis von Messergebnissen erfolgen. Hierbei sind für verschiedene charakteristische Tiefziehprozesse die Simulationsergebnisse den Ergebnissen von Messungen am realen Tiefziehteil gegenüberzustellen und die dabei auftretenden Differenzen durch die Wahl von besser geeigneten Simulationsmodellen und -parametern in einem iterativen Prozess zu minimieren.

Ein Abgleich zwischen Simulation und Realität ist auch dann notwendig, wenn in einem Tiefziehsimulationsprogramm neue Modelle, z.B. zur Beschreibung des Materialverhaltens, verwendet werden.

Neben dem Vergleich von Mess- und Simulationsergebnissen kann es auch erforderlich sein, verschiedene Simulationsergebnisse miteinander zu vergleichen. Diese ergeben sich aus der Variation von Simulationsmodellen oder -parametern bzw. aus der Verwendung verschiedener Simulationsprogramme. Gegenstand des Vergleichs sind dabei in der Regel die in Tiefziehprozessen auftretenden Formänderungen (Dehnungen), Blechdicken und Blechdickenänderungen sowie die bei der Rückfederung des Tiefziehteils auftretenden Verschiebungen.

Um den Vergleich zwischen Mess- und Simulationsergebnissen sowie zwischen Simulationsergebnissen untereinander durchführen zu können, benötigt man Software mit geeigneter Vergleichs- und Auswertefunktionalität. Nachfolgend werden die Anforderungen an eine solche tiefziehspezifische Vergleichs- und

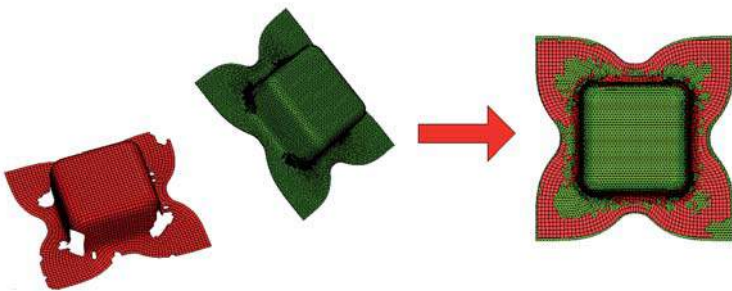


Abb. 1: Einschwimmen zweier Diskretisierungen eines Tiefziehteils zu Vergleichszwecken (vom Mess- bzw. Simulationssystem erzeugt).

News

➔ Animator4, Version 2.0.0 veröffentlicht

Die neue Animator4-Version bringt viele neue Funktionalitäten, mehr unter <http://www.gns-mbh.com>

➔ Knorr-Bremse Ungarn beauftragt GNS Systems mit CAE-Support

Beiträge

➔ Vergleich von tiefziehspezifischen Mess- und Simulationsdaten mit dem OpenForm-Postprozessor

➔ Tcl für Animator4 (A4), Teil II

➔ Anwenderbericht Faurecia: Standortübergreifende Software-Verteilung

➔ Unterstützung konstruktionspezifischer FATXML-Daten in Animator4



Auswertefunktionalität formuliert und deren Umsetzung im *OpenForm*-Postprozessor vorgestellt.

Anforderungen

Unter dem Gesichtspunkt praktischer Anwendbarkeit umfasst ein Vergleich von Mess- und Simulationsergebnissen weit mehr als nur das Berechnen und Visualisieren bestimmter Differenz-Ergebnisgrößen. In der Regel liegen die Ergebnisse von Mess- und Simulationssystemen für jeweils unterschiedlich diskretisierte Geometrien des Tiefziehteils vor. Sie unterscheiden sich auch bzgl. ihrer räumlichen Position voneinander. Um den Ergebnisvergleich durchführen zu können, muss die räumliche Position der zu vergleichenden diskretisierten Geometrien des Tiefziehteils zunächst in eine möglichst gute Übereinstimmung gebracht werden. Dieser Vorgang soll nachfolgend als Einschwimmen bezeichnet

werden (Abb. 1). In der Praxis benötigt man sowohl automatische als auch manuelle Methoden zum Einschwimmen zweier diskretisierter Geometrien eines Tiefziehteiles. Automatische Methoden berechnen die Koordinatentransformation für die einzuschwimmende Geometrie so, dass der Abstand zwischen den beiden aufeinander einzuschwimmenden Geometrien minimiert wird. Das manuelle Einschwimmen kann durch interaktives Positionieren der einzuschwimmenden Geometrie z.B. mittels Maus, durch das Definieren von Rotationen und Translationen oder durch das Definieren von jeweils drei Referenzpunkten auf einzuschwimmender Geometrie und Zielgeometrie erfolgen. Beim Einschwimmen ist außerdem zu beachten, dass die aufeinander einzuschwimmenden diskretisierten Geometrien unterschiedliche Flächen in Dickenrichtung des Tiefziehteils beschreiben

können, wie z.B. Mittelfläche, Außenfläche oder Innenfläche.

Nach dem Einschwimmen zweier diskretisierter Geometrien können aus den ihnen zugeordneten Ergebnisgrößen die Differenz-Ergebnisgrößen berechnet werden. Dabei ist zu gewährleisten, dass die Berechnung von Differenz-Ergebnisgrößen auch für zwei völlig unterschiedliche geometrische Diskretisierungen des Tiefziehteils möglich ist.

Da in der Praxis sowohl verschiedene Messsysteme (*Argus*, *AutoGrid*), als auch verschiedene Tiefziehsimulationssysteme (*AutoForm*, *PamStamp*, *LS-DYNA*, *INDEED*) eingesetzt werden, sollte ein Postprozessor mit integrierter Vergleichsfunktion über die entsprechenden Schnittstellen zum Einlesen der Ergebnisdaten dieser Systeme verfügen.

Realisierung

Die Realisierung der oben genannten Anforderungen erforderte eine Weiterentwicklung des *OpenForm*-Postprozessors in den folgenden Punkten:

- Implementierung eines Werkzeugs zum Einschwimmen von diskretisierten Geometrien
- Berechnung und Visualisierung von Differenz-Ergebnisvariablen und geometrischen Abständen zwischen zwei zu vergleichenden Geometrien
- Implementierung von Schnittstellen zu Messsystemen und Simulationsprogrammen

Das dem *OpenForm*-Postprozessor hinzugefügte Transform Process Tool (Abb. 2) ermöglicht nun das Einschwimmen von diskretisierten Geometrien des Tiefziehprozesses, wie z.B. Platinen, und das Transformieren kompletter Tiefziehprozesse unter Verwendung der beim Einschwimmen vorgenommenen Koordinatentransformation.

Zum automatischen Einschwimmen von Geometrien verwendet das Transform Process Tool den Iterative-Closest-Point(ICP)-Algorithmus. Der ICP-Algorithmus arbeitet auf der Grundlage von

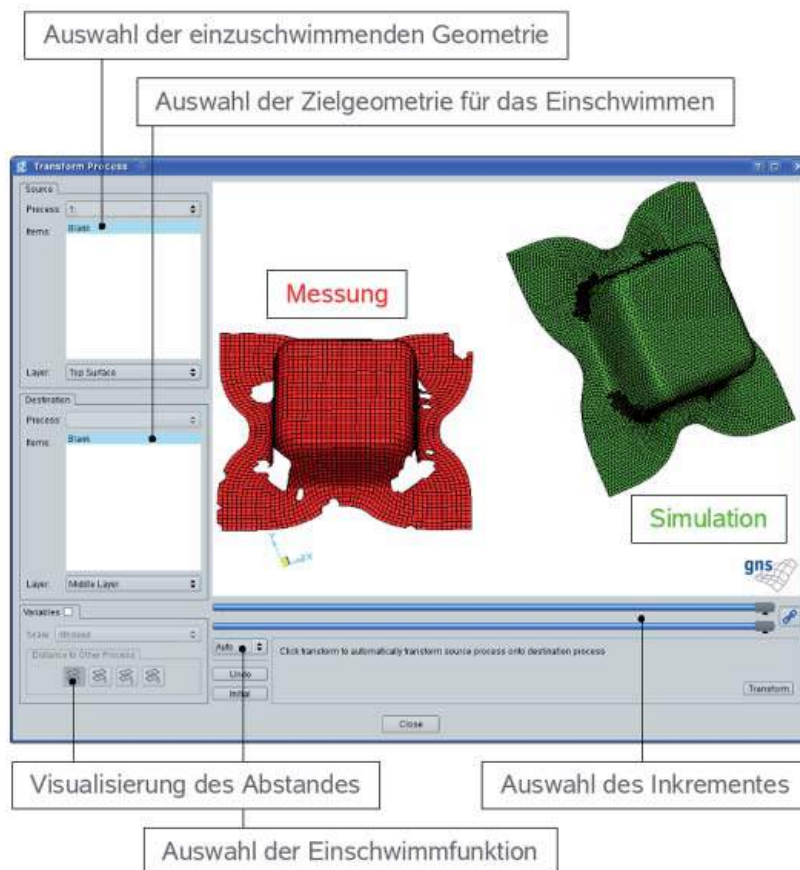


Abb. 2: *OpenForm* Transform Process Tool zum Einschwimmen von diskretisierten Geometrien

Punktwolken und ermöglicht es, diese aneinander anzupassen. Dazu werden für die Punktwolken Koordinatentransformationen so bestimmt, dass die Abstände zwischen den Punktwolken ein Minimum erreichen. Zunächst wird für jeden Punkt aus der einen Punktwolke der jeweils nächste Punkt aus der anderen Punktwolke ermittelt. Das Minimieren der Summe der Quadrate der Abstände

Weiterhin erlaubt es das Transform Process Tool, die Transformationsmatrix für eine gefundene Einschwimmposition in eine Datei zu exportieren bzw. von einer Datei zu importieren.

Neben der Einschwimmfunktion wurde der *OpenForm*-Postprozessor um Funktionen zum Berechnen und Visualisieren von Differenz-Ergebnisgrößen und

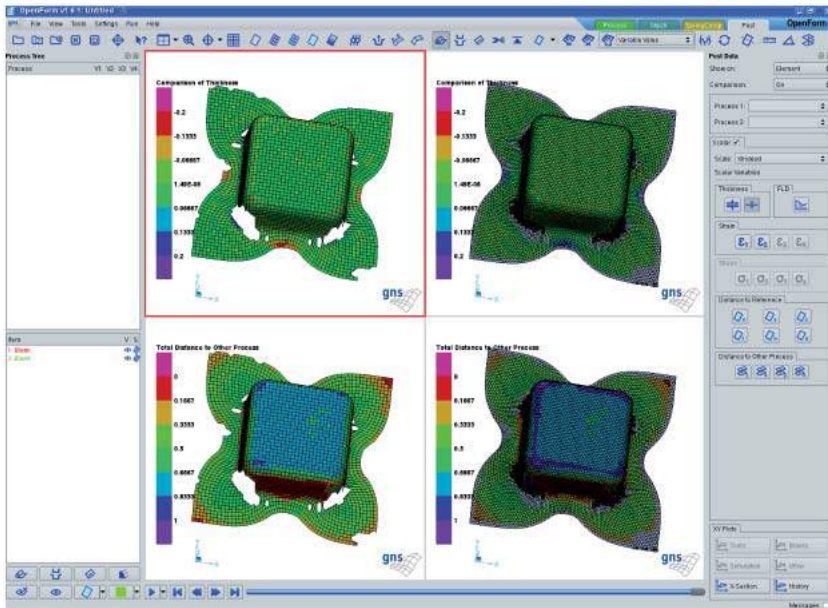


Abb. 3: Vergleich von Mess- und Simulationsergebnissen für ein Tiefziehteil mit OpenForm

erfolgt durch Anpassung der Transformationsparameter. Dieser Vorgang wird iterativ so lange wiederholt, bis das Optimum gefunden ist.

Für das manuelle Einschwimmen bietet das Transform Process Tool die folgenden Möglichkeiten zum interaktiven Positionieren der einzuschwimmenden Geometrie:

- Verschieben/Drehen mittels Maus
- Direkte Eingabe von Rotationen und Translationen
- Definieren von jeweils drei Referenzpunkten auf einzuschwimmender Geometrie und Zielgeometrie per Mausklick oder durch direkte Eingabe

geometrischen Abständen zwischen zwei zu vergleichenden Geometrien ergänzt (Abb. 3).

Dank einer umfangreichen Erweiterung der vom *OpenForm*-Postprozessor unterstützten Schnittstellen können jetzt Ergebnisse der Messsysteme *AutoGrid* und *Argus* sowie der für Tiefziehsimulationen eingesetzten Softwareprodukte *AutoForm* (ASCII Result Files), *INDEED*, *LS-DYNA* (d3plot Files), und *PamStamp* (Pam Mapping Files) visualisiert und miteinander verglichen werden.

Michael Wenzlaff, GNS mbH
openform@gns-mbh.com



Tcl für Animator4 (A4), Teil II

In der letzten Ausgabe des Genius wurden die Grundlagen der *Animator4*-Tcl-Schnittstelle vorgestellt. Hierbei wurden folgende Themen behandelt:

- Schleifen
- Parameterübergabe
- Graphische Oberflächen und Funktionen

In dieser Ausgabe wird eine Auswertung von Modelldaten demonstriert. Dabei werden Techniken verwendet, die aus dem ersten Teil bekannt sind:

- Schleifen und
- Parameterübergabe

Wie schon in der vorangegangenen Ausgabe können die umfangreichen Möglichkeiten hier nur angedeutet werden.

Die Beispiele dieser und der vorherigen Ausgabe können von der GNS-Homepage unter <http://gns-mbh.com/?id=GeniusTCL> geladen werden.

Zugriff auf Modelldaten

Dank der A4-Tcl-Schnittstelle können interne Informationen wie Modelldaten abgefragt und verarbeitet werden. In Beispiel 3 werden Informationen über Bauteile abgefragt. Das Beispiel-Skript soll nur Bauteile anzeigen, deren Dicke in einem definierten Bereich liegt.

Der obere und untere Grenzwert für die Bauteildicke werden beim Start als Parameter übergeben. Sind keine oder weniger als zwei Parameter angegeben, werden Standardwerte angenommen (Zeilen 1-6).

Tcl-Skripte können *Animator4*-Kommando-Targets berücksichtigen, die beim Aufruf des Skripts verwendet werden (z.B. `s[all]:rea tcl ...`); die Liste der zu bearbeitenden Slots ist dann in `cmdTargets(SLOTS)` abgespeichert. Diese Liste beinhaltet abwechselnd die Ids und Namen der Slots.

Die `foreach`-Schleife eignet sich ideal dazu derartige Listen abzuarbeiten. Die in Zeile 11 angegebene `foreach`-Anweisung extrahiert paarweise Werte aus der Slot-Target-Liste und weist diese für jeden Schleifendurchlauf den Variablen `slotId` und `slotName` zu. (Der `slotName` wird im Beispiel nicht weiter verwendet.) Für jede Variable `slotId` wird der Anweisungsblock der Schleife von Zeile 11 bis 53 abgearbeitet.

Innerhalb der Schleife wird für jede `slotId` eine Liste von Bauteilnummern extrahiert. Das geschieht mit dem Kommando `GNS_get_pid_list` (Zeile 16).

Bevor die Liste der Bauteile ausgewertet wird, werden in den Zeilen 18 und 20 zwei leere Listen erzeugt, welche im weiteren Verlauf mit Bauteilnummern (`addList`) sowie zusätzlich mit dem Bauteilnamen und der Dicke selbst (`tList`) gefüllt werden.

In der inneren `foreach`-Schleife (Zeilen 23-35) wird für jedes Bauteil aus der Bauteilliste mit Hilfe des A4-Tcl-Kommandos `GNS_get_property` ein Tcl-Array mit den Bauteileigenschaften erzeugt. Dies geschieht durch das geschachtelte Kommando in Zeile 25. Das innere Kommando liefert als Rückgabewert eine Liste, die in abwechselnder Reihenfolge Bezeichner und Werte von Bauteileigenschaften enthält. Aus dieser Liste wird durch das äußere Kommando `array set props` ein Tcl-Array mit den Namen `props` gebildet. Nun lassen sich die jeweiligen Eigenschaften mit Hilfe der Bezeichner als Schlüssel abfragen. So liefert z.B. `$props(THICK)` die Dicke des Bauteils.

Nachdem in Zeile 27 überprüft wird, ob das aktuelle Bauteil über eine Dickenangabe verfügt, wird in Zeile 29 die Dicke aus dem Array der Eigenschaften gelesen und in der Variable `t` gespeichert.

Die `if`-Abfrage in Zeile 32 wird benutzt, um nur die Bauteile zu berücksichtigen, deren Dicke in dem Bereich liegt, den der Anwender durch die Parameter definiert hat. Ist dies der Fall, wird die Bauteilnummer an die Liste `addList` angehängt. Zusätzlich werden die Dicke, die Bauteilnummer und die Bauteilbe-

zeichnung in der Liste `tList` gespeichert.

Nachdem alle Slots abgearbeitet sind, wird in Zeile 37 die Liste von Bauteilen in eine Zeichenkette umgewandelt, welche die Bauteilnummern in einer A4-konformen Listendefinition enthält und somit in A4-Kommandos eingesetzt werden kann. Die erstellte Zeichenkette wird anschließend in Zeile 42 verwendet.

Um die Bauteile einzublenden, müssen zuvor alle Bauteile mit dem A4-Kommando `era pid all` ausgeblendet werden. Dies darf jedoch nur für den Slot des aktuellen Schleifendurchlaufs erfolgen. Sollen mehrere A4-Kommandos abgesetzt werden, bietet sich die Verwendung des A4-Tcl-Kommandos `GNS_exe_command` unter Benutzung von geschweiften Klammern an, da dann die Kommandos, wie bei Session-Files bekannt, einfach untereinander geschrieben werden können. Tcl verhindert jedoch, dass innerhalb geschweiften Klammern Variablen durch ihren Wert ersetzt werden. Daher ist es notwendig, `GNS_exe_command` mit dem zusätzlichen Steuerwort `variables` zu verwenden. Dies bewirkt, dass die Werte der Variablen `slotId` und `addString` auch bei der Verwendung geschweiften Klammern eingesetzt werden.

Zusätzlich sollen alle eingblendeten Bauteile mit einem Text versehen werden, welcher den Bauteilnamen und die Dicke beinhaltet. Hierfür müssen zunächst alle vorhandenen Bauteiltexte gelöscht werden.

Obwohl in Zeile 45 nur ein Kommando abgesetzt wird, werden geschweifte Klammern verwendet. Dies ist notwendig, damit die von Tcl als Steuerzeichen interpretierten Klammern „[“ bzw. „]“ des Slot-Bezeichners direkt verwendet werden können. Da das Kommando die Variable `$slotId` enthält, wird wie zuvor das Steuerwort `variables` für das A4-Tcl-Kommando `GNS_exe_command` verwendet.

Anschließend wird mit Hilfe einer `foreach`-Anweisung über die Elemente der Liste `tList` iteriert. Hierbei werden

```
01. # set min. thickness: use 1st argument or 0.0 if not given
02. set min [expr { $argc < 1 ? 0.0 : $argv(1)}]
03. # set max. thickness: use 2nd argument or 1e10 if not given
04. set max [expr { $argc < 2 ? 1e10 : $argv(2)}]
05. # if exactly one argument given, use min for exact match
06. set max [expr { $argc == 1 ? $min : $max   }]
07.
08. # print message to inform the user about the input values
09. GNS_exe_comand "opt ech \"Showing pids with $max >= t >= $min\""
10. # loop over target slots
11. foreach {slotId slotName} $cmdTargets(SLOTS) {
12.     # give message which slot is accessed
13.     GNS_exe_command "opt ech Working on Slot $slotId: $slotName"
14.
15.     # use GNS_get_pid_list to get a list of pids
16.     set pidList [ GNS_get_pid_list $slotId mat "*" ]
17.     # create empty list to hold valid pids
18.     set addList [list]
19.     # create empty list to hold valid pids with thick. and name
20.     set tList [list]
21.
22.     # loop over list of pids
23.     foreach pid $pidList {
24.         # get array holding the properties of the current pid
25.         array set props [GNS_get_property $slotId $pid]
26.         # check if thickness is given
27.         if { [ array names props THICK ] != "" } {
28.             # extract thickness from property array
29.             set t $props(THICK)
30.
31.             # append pid and props to lists if thickness is in range
32.             if { $t <= $max && $t >= $min } {
33.                 lappend addList $pid
34.                 lappend tList $pid $t $props(NAME)
35.             } } }
36.     # create suitable A4 list representation using slashes
37.     set addString [format "%s" [string map {{ } /} $addList]]
38.
39.     # erase all pids and add pids with thickness in range
40.     GNS_exe_command variables {
41.         s[$slotId]:era pid all
42.         s[$slotId]:add pid $addString
43.     }
44.     # delete any texts previously attached to pids
45.     GNS_exe_command variables {s[$slotId]:txt pid del all}
46.     # extract items from tList
47.     foreach {pid t name} $tList {
48.         # set string to be attached to pid
49.         set tString [format "%s: t=%2.2f" $name $t]
50.         # attach text to pid
51.         GNS_exe_command variables {
52.             s[$slotId]:txt pid add $pid "$tString"
53.         } } }
```

Beispiel 1: Einblenden und Beschriften von Bauteilen einer bestimmten Dicke (thickness.tcl)

bei jedem Durchlauf die Bauteilnummer sowie die Dicke und der Name ausgelesen. Der Name und die Dicke werden zu einer Zeichenkette formatiert, welche anschließend mit dem A4-Kommando `txt add pid` an das jeweilige Bauteil gehängt werden.

Fazit

Fallunterscheidungen und Schleifen auf der Grundlage von Ergebnisdaten eröffnen ganz neue Möglichkeiten; die Tcl-Schnittstelle von *Animator4* ermöglicht damit die Automatisierung von äußerst komplexen Auswertungen. Das erspart den Anwendern monotone Arbeitsabläufe und sichert gleichzeitig einheitliche Rahmenbedingungen für vergleichbare Ergebnisse.

*Bernd Antes, GNS mbH,
Sascha Kremers, GNS mbH,
Björn Brodersen,
animator@gns-mbh.de*



Anwenderbericht Faurecia: Standortübergreifende Software-Verteilung

SAFE ist eine Lösung zur standardisierten CAD/CAE-Softwareverteilung beim Automobilzulieferer Faurecia. Entstanden ist SAFE in der langjährigen Zusammenarbeit mit GNS Systems. Ziel von Faurecia war es, die IT-Administration zu optimieren und dabei die Verwaltungskosten für die Ingenieursarbeitsplätze zu senken.

Während es für die Verteilung von Büro-Applikationen und betriebswirtschaftlicher Software zahlreiche Lösungen im Markt gibt, ist bei der Ausstattung von CAD- und CAE-Arbeitsplätzen mit Software oft noch viel Handarbeit erforderlich. Die Thematik verschärft sich, wenn sie – wie beim Automobilzulieferer Faurecia – die Konstruktions- und Entwicklungsarbeitsplätze an fast 70 Standorten weltweit betrifft.

Unternehmen sind heute nicht mehr eine starre Masse, sondern wandeln sich stetig. Wenn die CAD/CAE-Arbeitsplätze neuer Standorte in das Unternehmen hinein wachsen, geht das stets über die Hardware- und Software-Themen hinaus. Unterschiedliche Arbeitsmethoden, Kulturen und Sprachen müssen gemeistert werden. Einheitlich ausgestattete CAD/

CAE-Arbeitsplätze sind besonders wichtig, wenn der Vorteil unterschiedlicher Denk- und Lösungsansätze ausgeschöpft werden soll. Dafür muss das Softwarepaket umso mehr für alle Administratoren gleich gestaltet und gleich verständlich sein.

„Unser Entwickler-Team ist international. 15 Nationalitäten arbeiten in unterschiedlichen Zeitzonen. Da treffen auch kulturelle Unterschiede aufeinander. So spielen die Weihnachtsfeiertage an unseren europäischen Standorten eine große, in Indien jedoch eine eher untergeordnete Rolle“, erklärt Wolfgang Schwegel, der bei Faurecia für den weltweiten Betrieb der CAD/CAE-Arbeitsplätze verantwortlich ist.

IT-Landschaft „im Fluss“

Faurecia ist Teil des PSA-Konzerns mit den Marken Peugeot und Citroën und hat alle namhaften Automobil-OEMs als Kunden. Daher müssen die jeweils beim Kunden eingesetzten CAX/PDM-Systeme auch an den verschiedenen Faurecia-Standorten verfügbar sein.

Rund 200 CAX-Softwarepakete müssen an den Standorten zur Verfügung gestellt werden. Dabei sind unterschiedliche Betriebssysteme und Hardware-Modelle zu berücksichtigen. Das ist die tägliche Herausforderung von Wolfgang

Schwegel mit seinem Team. Seit einigen Jahren stützt sich Faurecia dabei auch auf die Erfahrungen und das Programmier-Know-how des IT-Dienstleisters GNS Systems aus Braunschweig.

Schwegel erläutert: „Zu Beginn unserer Zusammenarbeit mit GNS Systems waren allein 16 verschiedene Versionen der Engineering-Software CATIA V4 im Einsatz. Andere Standorte entwickelten bereits mit CATIA V5-Versionen.“ Auch Unigraphics und Pro/Engineer gehören zur Software-Vielfalt. Enovia ist das unternehmensweit genutzte PDM-System. Zusammen mit CAE-Programmen wie LS-DYNA, Abaqus, StarCD, GT-Power und Hyperworks kommt Schwegel auf 200 Applikationen und Softwareversionen, zu denen auch die spezifischen Konstruktionslösungen entsprechend der Kundenaufträge gehören. „Unsere IT-Landschaft ist immer im Fluss“, sagt Schwegel: „Wenn unsere Kunden CATIA V5 – auch in unterschiedlichen Versionen – einsetzen, müssen wir das im Entwicklungsprojekt berücksichtigen.“

Die Architekturdefinition Arche für die CAD/CAE-Applikationslandschaft wurde ursprünglich von Faurecia anhand der Vorgaben aus dem Hause PSA erarbeitet und wurde über die Jahre weiter entwickelt. Wie genau der Begriff Arche entstand, lässt sich nicht mehr so richtig nachvollziehen. Manche Faurecia-Mitarbeiter sehen darin den großen, alles überspannenden Bogen, andere sehen eine Analogie zum biblischen Schiff, denn auch in der Software-Beschreibung sollen von jedem CAX/PDM-Programm nur eine bestimmte Art und Version zugelassen sein. Jedoch gab es – das liegt in der Natur der Sache – an jedem Standort ganz unterschiedliche Auffassungen über die Art des Roll-outs der Softwarepakete. Schwegel: „Es sind ganz einfache Ursachen, die das Roll-out der Software behindern können. Während in einem Land vielleicht das Laufwerk Y für TEMP-Dateien verwendet wird, ist Y an einem anderen Standort für den Backup-Datenträger reserviert. Zudem müssen wir davon ausgehen, dass nicht überall die gleichen Fachkenntnisse vorhanden sind.“



Abb. 1: Mit der Referenzarchitektur wird die Synchronisation der Engineering-Software grundsätzlich geregelt.

Vereinfachung durch Standardisierung

SAFE, Same Arche For Engineers, war deshalb das Ziel von Wolfgang Schwegel. Weil Administration und Programmierung im CAD/CAE-Umfeld keine triviale Aufgaben sind und weil die IT-Verantwortlichen an vielen *Faurecia*-Standorten auch für die Anwenderbetreuung und den PC-Support verantwortlich sind, benötigte der Automobilzulieferer entweder eine im Markt verfügbare Komplettlösung oder einen Partner für die Entwicklung der Skripte und Systemsoftware.

„Wir haben im Markt kein kommerzielles oder gar lizenzfreies Produkt gefunden, das unseren Vorstellungen entsprach.“ Bereits seit einigen Jahren ist der Arbeitsmarkt für Software-Entwickler wie leergefegt, so dass auch die Eigenentwicklung für *Faurecia* keine taugliche Option war.

Mit *Animator4*, einer Software zur Visualisierung von FEM-Berechnungen von *GNS - Gesellschaft für numerische Simulation mbH* hatte *Faurecia* bereits Erfahrungen gesammelt. So wurde schließlich das Schwesterunternehmen *GNS Systems* mit der Programmierung einer standardisierten Software-Verteilung beauftragt.

Software-Standardisierung sichern

„Unser Motiv für *SAFE* war die Standardisierung und eine daraus hervorgehende Automatisierung der Softwareverteilung für die CAD- und CAE-Workstations“, so Schwegel. In einer Bestandsanalyse, die *GNS Systems* gemeinsam mit der IT-Mannschaft von *Faurecia* vornahm, gewannen die Partner die Erkenntnis, dass die Architekturbeschreibung *Arche* durchaus als optimal angesehen werden kann. Um dem ständigen Release-Wechsel an den Standorten zu begegnen, wurde mit den Experten von *GNS Systems* eine offene Diskussion geführt. „Das Ergebnis war das Konzept für *SAFE*“, berichtet Schwegel.

Schrittweise – zunächst für die Konstruktionslösungen an den Standorten Stadthagen und Wolfsburg, später für alle Konstruktors- und Simulationsarbeitsplätze weltweit – wurde *SAFE* von

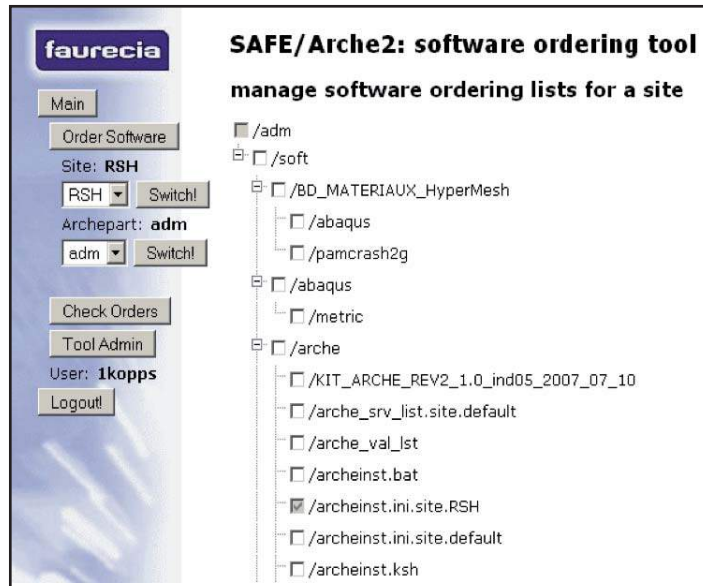


Abb. 2: Ein Element von *SAFE* ist das Ordering-Tool, das mit dem Lizenz-Server kommuniziert und hilft, die TCO der CAE-Umgebung zu reduzieren.

GNS Systems programmiert, getestet und als Software-Tool für die Administration der CAD- und CAE-Applikationen implementiert.

Die Standardisierung und Automatisierung von Software-Roll-outs ist eine diffizile Aufgabe. Schwegel: „Bei Schwierigkeiten hat *GNS Systems* jedoch sofort reagiert. Durch die örtliche Nähe und durch die detaillierten Kenntnisse unserer IT-Umgebungen konnten Störungen kurzfristig behoben werden.“



Abb. 3: Same Arche For Engineers ist die Lösung für die Software-Distribution für die weltweit verteilten Engineering-Arbeitsplätze bei *Faurecia*.

Kostensenkung: willkommener Effekt

SAFE hat die Software-Architektur für die *Faurecia*-Entwickler deutlich optimiert. An allen Standorten sind jetzt die gleichen Software-Releases für alle CAD- und CAE-Programme im Einsatz. Und *SAFE* hat einen willkommenen Nebeneffekt. „Wir haben mit *SAFE* auch das Lizenzmanagement verbessert“, erklärt

Wolfgang Schwegel und fügt hinzu: „Die Ausstattung eines Simulationsarbeitsplatzes kann schnell hohe Lizenzkosten verursachen.“

Die CAX-Administratoren an den Standorten können nun aus einem Menü vorgegebener Software-Releases wählen und erhalten in wenigen Minuten ein Installationspaket. Mit dem Installationspaket werden die veralteten Versionen gelöscht. Im Rahmen kontinuierlicher Zusammenarbeit erweitert *GNS Systems* die Skripte und Prozeduren in *SAFE* für alle aktuellen und künftigen Software-Releases der CAD- und CAE-Applikationen.

„Mit *SAFE* kontrollieren wir unsere Standorte gemäß *ITIL*“, resümiert Schwegel. Doch die nächsten Aufgaben stehen bereits im Raum: Schon haben viele Ingenieure den Wunsch nach mobilen Endgeräten geäußert. Hinsichtlich der komplexen CAD/CAE-Systeme wird es nicht leicht, jeden Wunsch zu erfüllen. Die Integration der Ingenieursarbeitsplätze mit der Welt der betriebswirtschaftlichen Software und der Büro-Applikationen soll ebenfalls gemeinsam mit *GNS Systems* vorangetrieben werden.

*Holm Landrock, freier Journalist
christopher.woll@gns-systems.de*

Termine

→ LS-DYNA Forum 2012

Datum: 09.-10. Oktober 2012
Veranstaltungsort: Ulm

Aussteller: GNS Systems GmbH
und GNS mbH

→ SIMVEC 2012

Berechnung, Simulation und
Erprobung im Fahrzeugbau

Datum: 20.-21. November 2012
Veranstaltungsort: Baden-Baden

Aussteller: GNS Systems GmbH
und GNS mbH

Weitere Termine finden Sie unter:
www.gns-mbh.com und
www.gns-systems.de



GNS mbH

Am Gaußberg 2
38114 Braunschweig
Telefon: 05 31-8 01 12 0
Fax: 05 31-8 01 12 79
www.gns-mbh.com



GNS Systems GmbH

Am Gaußberg 2
38114 Braunschweig
Telefon: 05 31-1 23 87 0
Fax: 05 31-1 23 87 11
www.gns-systems.de

Impressum

Ausgabe 1/2012
Erscheinungstermin: Juni 2012
Herausgeber: GNS Systems GmbH

Verantwortlich: Jan Martini
Redaktion: Anette Tröger
Layout: Anette Tröger

Alle Rechte vorbehalten.
Vervielfältigung, auch auszugsweise, nur mit
schriftlicher Genehmigung des Herausgebers.
Alle Markennamen sind in der Regel eingetragene
Warenzeichen der entsprechenden Hersteller
oder Organisationen.

Unterstützung konstruktions- spezifischer FATXML-Daten in Animator4

Mit dem FATXML-Format gibt es jetzt ein neutrales, standardisiertes Datenformat, mit dem konstruktionspezifische Daten in der gesamten CAE-Prozesskette bereitgestellt werden können. Die auf XML basierenden Datenbeschreibungen werden hierbei von den jeweiligen Präprozessoren erstellt und stehen bis in die Ergebnisdatei des Solvers zur Verfügung. Weiterführende Informationen zu FATXML finden sich unter [1].

Animator4 unterstützt die durch FATXML-Daten vorliegenden Konstruktionsdaten mit Hilfe von Layern. Layer ermöglichen eine hierarchisch strukturierte Sicht auf die Modelldaten. Außerdem können Layer beschreibende Daten („Meta-Daten“) enthalten, die sowohl Informationen über den Layer selbst als auch über die enthaltenen Properties geben. Gängige „Meta-Daten“ sind z.B. Blechdicken oder Materialnamen.

Erkennt *Animator4* beim Import einer Datei konstruktionspezifische Daten im FATXML-Format, so werden diese automatisch mitgelesen und auf unterschiedliche Arten präsentiert:

1. Der Layerdialog stellt die konstruktionspezifischen Daten in einer Baumstruktur dar. Zugehörige Meta-Daten werden als Tooltips in einem eigenen Fenster angezeigt, sobald ein Layer, der Zusatzinformationen besitzt, mit der Maus überfahren wird. Dies ermöglicht einen schnellen Überblick über die Daten.

2. Mit der *Animator4*-Version 2.0.0 stehen die Meta-Daten zusätzlich nach Anwendung des Kommandos „ide lay ...“ als Ergebnisvariablen zur Verfügung. Die erzeugten Variablen findet der Nutzer im

Variablendialog und kann sie verwenden, um z.B. Bildschirmtexte zu erzeugen.

3. Den umfangreichsten und flexibelsten Zugriff bietet die Möglichkeit, die Meta-Daten mit Hilfe der *Animator4*-eigenen Tcl-Schnittstelle abzufragen und auszuwerten (siehe Beispiel unten).

Die Verfügbarkeit der Meta-Informationen in der Tcl-Schnittstelle bietet eine Vielzahl von Möglichkeiten zur Auswertung der Daten. Z.B. können Blechdicken aus Konstruktion und Berechnung verglichen und Unterschiede aufgezeigt werden.

FATXML-Daten werden zurzeit von *Animator4* nur in *Pamcrash*-Dateien interpretiert, wobei die FATXML-Daten im Inputdeck und im ERF-Ausgabeformat verfügbar sein können. Die Unterstützung anderer Formate, wie *Abaqus* und *LS-Dyna*, ist geplant.

```
# Layer abfragen
array set layer [GNS_get_layer 0 "layername"]
# Ausgabe aller Attribute des Layers. Attribute
# können Property-spezifisch sein, in diesen Fällen
# enthält $pid die Property-Id
foreach {key value pid} $layer(ATTRIBUTES) {
    GNS_exe_command "opt ech $key $value $pid"
}
```

Beispiel: Ausgabe aller Layer-Attribute mit Tcl

Die Einbeziehung konstruktionspezifischer Daten in die gesamte CAE-Prozesskette ermöglicht eine hierarchisch strukturierte Sicht auf die Ergebnisdaten, die eine Auswertung der Daten erheblich erleichtert. Gleichzeitig wird eine Rückkopplung von der Berechnung zur Konstruktion unterstützt, so dass an dieser Stelle ein iterativer Prozess unterstützt wird.

Sven Trentmann
animator@gns-mbh.com

[1] <http://www.vda.de/de/arbeitsgebiete/FATXML/>