**Duchlaufträger Applikation Kurzdokumentation**

**Button**:  
enthalten "Tooltips", d.h. Kurzerklärung der Funktion, wenn Zeiger dorthin bewegt wird

**Menü**:  
**NEU** Dialog für einen neuen Träger, löscht vorhandene Ausgabe

**Einspannung** Dialog zum Ändern der Randbedingungen am Anfang und am Ende

**Lager** Dialog zur Definition/Editieren/Löschen eines Lager

**Einzellast** Dialog zur Definition/Editieren/Löschen einer Einzellast

**Gleichlast** Dialog zur Definition/Editieren/Löschen einer Gleichlast

jede Modelländerung (Randbedingungen, Lager, Last) führt zu einer direkten Neuberechnung und grafischen Darstellung der Ergebnisse

**Neuberechnung** explizite Veranlassung einer Neuberechnung mit aktuellen Modelldaten

**Übertragungs-** darstellen/abschalten aller Übertragungspunkte mit Nummer und  **punkte** Symbol (kleiner schwarzer Kreis)

**Momententexte** darstellen/abschalten aller Momententexte

**Querkrafttexte** darstellen/abschalten aller Querkrafttexte

Die grafischen Modell- und Ergebnisdarstellungen können interaktiv abgefragt und editiert werden. Generell wird der Träger auf die Fenstergröße skaliert und durch eine schwarze Linie dargestellt.

**Übertragungspunkte** werden durch eine Nummer und einen kleinen blauen Kreis markiert und identifiziert. Ein Linkssclick auf die Nummer öffnet ein blaues Pop-Up Fenster am Ort des Zeigers mit dem aktuellen Zustand des Übertragungspunktes. Dieser Zustand beinhaltet den Identifikator (Nummer) des Übertragungspunktes und die Verschiebung, Verdrehung, Biegemoment und Querkraft links und rechts des Übertragungspunktes. Ein Rechtsclick auf die Nummer schließt dieses Pop-Up Fenster wieder.

Direkt unterhalb des Trägers werden die Positionen (Längen) der Übertragungspunkte auf dem Träger unter den Identifikationsnummern als schwarzer Text angezeigt.

Unterhalb der Trägerstellung wird der Biegemomentenverlauf dargestellt mit den Ergebnistexten in rot. Darunter wird der Querkraftverlauf dargestellt mit den Ergebnistexten in blau.

Ein **Linksclick auf Modellobjekte (Lager, Lasten und Randbedingungen)** öffnet die entsprechenden Dialogmasken zum Editieren oder Löschen des jeweiligen Modellobjektes.

Speziell das Editieren einer Einzellast wird anschaulich unterstützt durch die Darstellung eines roten (Pilot-)Punktes am Lastangriffspunkt. Wird der Zeiger auf den Pilotpunkt geführt, wird dieser Punkt aktiviert und kann durch (langsames) Bewegen des Zeigers auf dem Träger interaktiv horizontal verschoben werden. Gleichzeitig wird die aktuelle Koordinatenposition auf dem Träger im Dialogfenster kontinuierlich aktualisiert und mitgeführt. Ein Rechtsclick deaktiviert die interaktive Positionsänderung des Pilotpunktes und die zugehörige Koordinatenanzeige kann noch editiert werden und/oder durch OK akzeptiert werden.

Jede Modelländerung führt zu einer unmittelbaren Neuberechnung und Ergenisdarstellung.

**Berechnungsergebnisse** werden grafisch und als Ergebnistexte an den Übertragungspunkten dargestellt.

Speziell die Informationen der Ergebnisse der Momentenverläufe unter Gleichlasten (quadratische Exponentialfunktionen) wird noch unterstützt durch interaktive Abfrage der Maximalmomente, da standardmäßig nur die Werte an den Übertragungspunkten ausgegeben werden. Hierzu können mit einem Linksclick auf den Verlauf der Biegemomentenkurve die Maximalmomente am Ort ihres Auftretens ausgegeben bzw. wieder ausgeschaltet werden.

**Erstes Fazit der Umsetzung einer Ingenieurapplikation in C#, .NET**

Aus Sicht der Informatik ergibt sich mit den Möglichkeiten der Anwendungsentwicklung in dieser Entwicklungsumgebung grundsätzlich nichts Neues an Möglichkeiten und Funktionalitäten. Alles, was hier implementiert wurde, ließ sich auch schon in der „alten“ Entwicklungsumgebung von *C++, Windows Forms* umsetzen. Allerdings gestehen auch Informatiker vielfach den resultierenden Ergebnisse zu, dass diese oft Code produzieren, der kunstvoll erstellt wurde, schwierig bis kaum nachvollziehbar ist und sich einem Dritten nur sehr schwierig erschließt. Diese Tatsache hat sogar in kommerziellen, sogenannten „legacy“ Anwendungen dazu geführt, dass bestimmte Code-Bestandteile nach Jahren ihrer Erstellung nicht mehr angetastet werden. So ist z.B. von dem ursprünglichen, alten AutoCAD bekannt, dass hocheffiziente, funktionierende, zentrale Bestandteile der Kernimplementierung nicht mehr angetastet wurden, da ihre Umsetzung so kunstvoll und kryptisch war, dass sie den aktuellen Entwicklern nicht mehr bekannt war und sich ihnen auch nicht mehr erschlossen hat („never touch a running system“).

Aus Sicht der Ingenieure ist dies nicht akzeptabel. Computercode für Ingenieuranwendungen muss so gestaltet sein, dass Lösungsalgorithmen problemlos erkennbar und nachvollziehbar sind und dass Erweiterungen und neue Verfahren als Ergebnisse der Forschung von Ingenieuren transparent implementiert und in bestehende Lösungen integriert werden können. Dabei ist es jedoch unverzichtbar und muss unbedingt gewährleistet bleiben, dass moderne Ingenieuranwendungen an den Ergebnissen der Forschung und Entwicklung in der Informatik partizipieren und an aktuellen Ergebnissen teilhaben.

Der schiere Umfang an Daten in modernen Ingenieuranwendungen, die transient im Kernspeicher moderner Rechner oder persistent in externen Speichermedien gespeichert und verwaltet werden, war vor wenigen Jahren noch unvorstellbar und verbietet es Ingenieuren mit alt herkömmlichen Verfahren auf diese Datenmengen zuzugreifen und diese zu verwalten. Frühe Implementierungen der FE-Methode haben z.B. die Speicherverwaltung noch selber mit Integer-Pointern in einem Blank-Common Block in FORTRAN implementiert. Dies ist heute unvorstellbar.

Allein das „Aufräumen“ der Speicher nach dem Ablauf komplexer Programmsystem („garbage collection“) ist heute eine anspruchsvolle Aufgabe, die verhindern soll, dass der verfügbare Speicherplatz scheinbar kontinuierlich abnimmt („memory leakage“). Aktuelle Ingenieurapplikationen sind meist unzureichend geschützt vor diesen Problemen.

Die allgemeine und selbstverständliche Nutzung mehrerer, hochkomplexer, paralleler Rechenkerne sogar in marktüblichen Laptops geht heute weit über die Vorstellungen eines linearen Programmablaufs hinaus und muss entsprechend organisiert und synchronisiert werden.

Die Anforderungen an die Erscheinungsform und Funktionalitäten der Benutzeroberflächen moderner Programmsysteme (UI) werden heute selbstverständlich und in standardisierter Form gefordert. Hierzu wurde als neue Schnittstelle die „Windows Presentation Foundation - WPF“geschaffen. Da dies eine interpretive Schnittstelle ist, kann diese relativ leicht und dynamisch durch die Nutzer als Benutzerschnittstelle genutzt werden. Ein besonderer Aspekt von WPF ist die unglaubliche Tiefe und Vielfalt vordefinierter Ereignisse, die standardmäßig genutzt werden können ohne speziell für jeden Einzelfall spezifisch erstellt werden zu müssen.

Aus meiner Sicht ist es die Aufgabe der Informatik, diese Aufgabenstellungen anwendungsübergreifend zu lösen und die Aufgabe der Ingenieure, die aktuellen Ergebnisse der Forschung und Entwicklung in der Informatik für moderne Ingenieuranwendungen zu nutzen.

Ebenso bedarf es aus meiner Sicht für diesen Prozess immer noch einen Transformationsmechanismus, der die Übertragung der Ergebnisse der Informatik in die Anwendungsdisziplinen leistet und anwendbar macht. Im Bauingenieurwesen ist dies die Bauinformatik, die sicherstellen muss, dass die aktuellen Entwicklungen und Konzepte der Informatik in Anwendungsentwicklungen im Bauingenieurwesen umgesetzt und genutzt werden können.

Die Informatik wird und kann in der Regel auch keine Forschung und Entwicklung von Bauingenieurapplikationen übernehmen.

Der allgemeine Bauingenieur wird neue Lösungsverfahren in seinem Fachgebiet entwickeln und bestehende weiterentwickeln. Eine herkömmliche „Programmierung“ dieser Ergebnisse ist heute den Anforderungen nicht mehr angemessen und kann kaum moderne Entwicklungen und Forschungsergebnisse der Informatik umsetzen und nutzen.

Die Umsetzung neuer Lösungsverfahren in moderne Ingenieurapplikationen muss von spezialisierten Fachleuten auf beiden Gebieten, der Informatik und der Anwendung, erfolgen, Eine reine Auftragsvergabe an Spezialisten wird den heutigen Anforderungen nicht mehr gerecht und eine kooperative Entwicklung von Fachleuten verschiedener Disziplinen gestaltet sich fast immer sehr schwierig.

Im Prinzip verschiebt sich die Grenze zwischen Systemsoftware und Anwendungsprogrammierung seit Jahren kontinuierlich. War es anfangs noch erstrebenswert „open software, open source“ für Betriebssysteme zu fordern und zu nutzen, ist dies schon seit einigen Jahren in vielen Anwendungsgebieten weder sinnvoll noch möglich. Ähnlich gestaltet sich die Entwicklung zwischen Betriebssystem und Anwendungsprogrammierung. Haben die Anwendungsprogrammierer anfangs noch Konzepte bis zur Programmierung auf Pozessorebene (Assembler) genutzt, ist dies heute weder sinnvoll noch nötig. Heute stehen extrem leistungsfähige Compiler zur Verfügung, die weit über die Möglichkeiten normaler Programmierer für Ingenieurapplikationen hinaus gehen. Inzwischen hat sich die die Grenze bis zur Nutzung einer sogenannten „Middleware“ verschoben, die viele moderne Konzepte der Informatik auf der Ebene der Anwendungsprogrammierung in einer modernen Entwicklungsumgebung anbietet. Damit hat sich die Grenze zwischen geschlossener Systemsoftware und offener Anwendungssoftware wieder ein Stück weit verschoben.

Die Kehrseite sind bei dieser Entwicklung die wachsenden Abhängigkeiten von Herstellern entsprechender Systeme - proprietäre Systeme, die eigentumsrechtlich einem bestimmten Anbieter gehören und in ihren inneren Zusammenhängen verborgen und nicht nachvollziehbar sind.

Die EU ist in dieser Entwicklung einen eigenen Weg gegangen. Dort wurde akzeptiert, dass die Software geschlossen und eigentumsrechtlich geschützt ist, aber es wurde die Forderung durchgesetzt, dass die **Anwenderdaten**, die mit diesen Systemen erzeugt werden offen sein müssen. Dies erfolgt z.B. im lesbaren Datenformat XML. Die Datenformate *docx* und *xlsx* der Firma Microsoft sind hierfür 2 Beispiele. Diese konnten nur mit hohem politischen Druck durchgesetzt werden. Andererseits waren die Argumente gegen digitale Versionen rechtlich verbindlicher Dokumente, die nur in einem intransparenten, eigentumsrechtlich geschützten Datenformat einer bestimmten Firma verfügbar waren, unwiderlegbar und inakzeptabel.