

Anwendungs-Programmsystem DAPROA

DAten- und PROzessAnalyse

Untersuchung (Auswertung, Analyse) experimenteller Daten,
Parameterschätzung, Modellbildung und -simulation

Version 5 .1

Stand: 31.10.2011

DAPROA

Das Programmsystem DAPROA umfasst Module(Programmbausteine), die ausgewählte Behandlungsverfahren und mathematische Methoden zur Prozess- und Datenanalyse, zur Auswertung und Analyse von Daten, Parameterschätzung in mathematischen Modellen, sowie Modellbildung und -simulation unterstützen.

Zielrichtung von DAPROA ist es Verfahren und Methoden bereitzustellen für

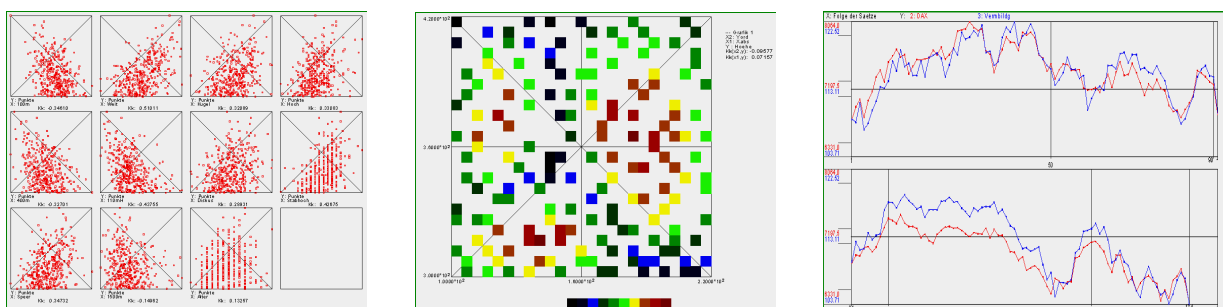
- die Untersuchung, Auswertung, Analyse von Daten aus geplanten und nicht geplanten Experimenten, Prozessen, Statistiken oder anderen Datenerhebungen
- die Schätzung von Parametern für lineare und nichtlineare mathematische Modelle
- die Berechnung und Simulation von mathematischen Modellen und Funktionen
- grafische Darstellung der Daten, sowie der Ergebnisse von Datenanalyse, Modellbildung und -simulation

Die auszuwertenden Daten werden DAPROA in einer standardisierten Textdatei übergeben. Diese Datei ist die Originaldatei, sie kann mit einem Texteditor leicht erstellt oder durch Entladen aus z.B. Office-Systemen oder Datenbanken erzeugt werden. Auch hier ist eine eventuelle Nachbearbeitung leicht möglich. Zu jeder so entstandenen Start-Datenmenge wird ein Projekt angelegt. Innerhalb dieses Projektes wird gearbeitet. Im Projekt können Unterprojekte entstehen, aber es gibt nur eine originale Datendatei. Die Dialogbildschirme und Grafiken sind an eine Auflösung von 1280x1024 Pixel angepasst.

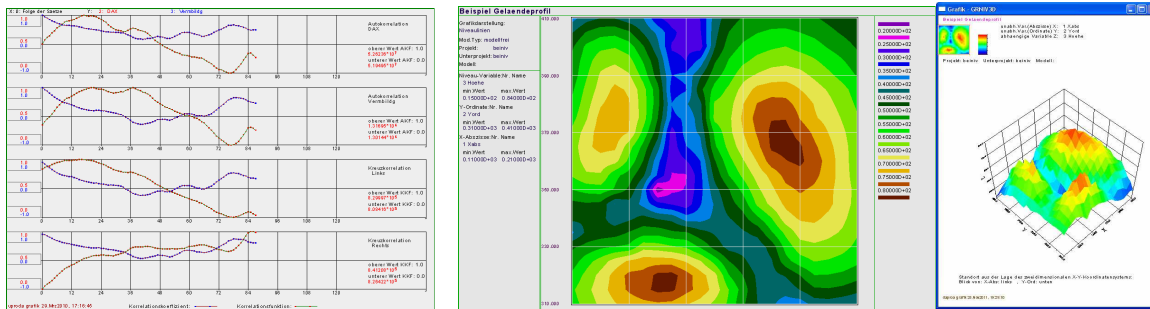
Was kann DAPROA?

Die hier beispielhaft gezeigten Grafiken sind natürlich ein Ausdruck der Ergebnisse der Datenbearbeitung mittels Verfahren und Methoden durch die Bausteine.

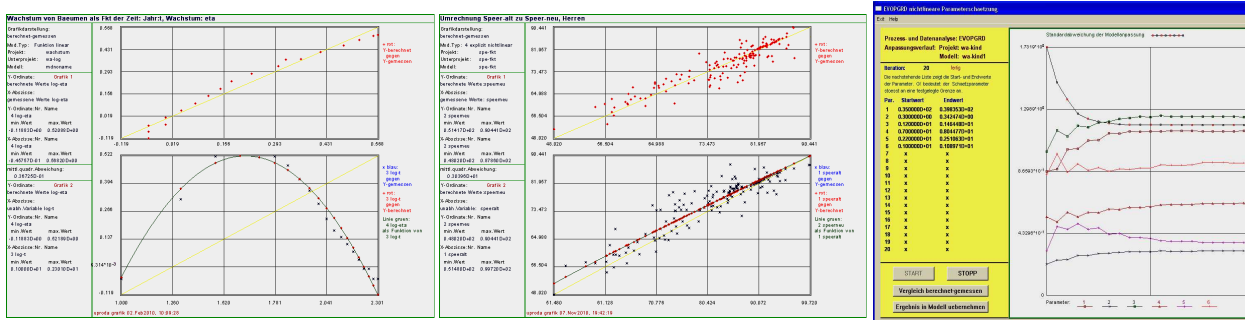
Grafiken aus den unbehandelten Daten



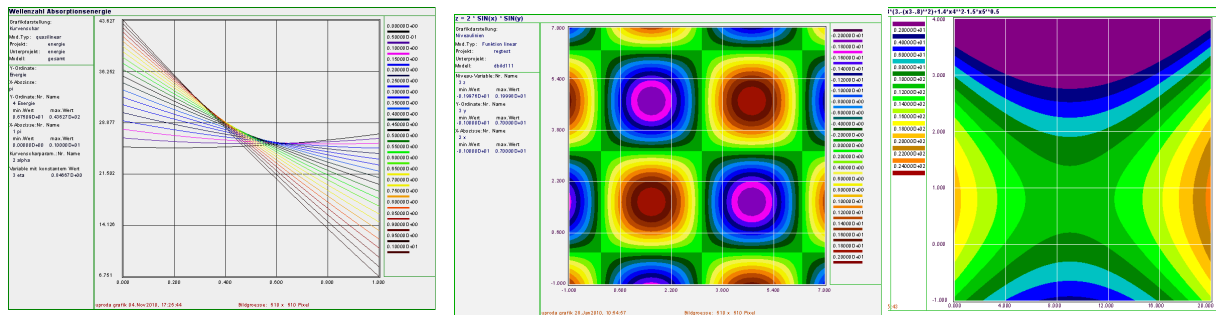
Grafiken aus mathematisch aufbereiteten Daten



Modellanpassung: Schätzung linearer und nichtlinearer Parameter mathem. Modelle



Auswertung und Simulation von mathematischen Modellen



Die Philosophie von DAPROA

DAPROA besteht aus Verarbeitungs- und Dialog-Programmen(.exe), und lesbaren Textdateien. Jeder DAPROA-Baustein ist ein ausführbares Programm vom Typ .exe. Die Module, Bausteine von DAPROA, arbeiten selbstständig voneinander. Sie korrespondieren nur über Dateien. Im praktischen Betrieb setzt die Abarbeitung bestimmter Module die Ergebnisse anderer Module voraus. Die Steuerung wird über die vorhandene Dialogoberfläche vorgenommen. Die DAPROA-Module laufen dabei im Hintergrund ab. DAPROA ist ein freies System und nutzt nur die Betriebssystemebene. Es baut keine internen Strukturen auf, in die der Anwender keinen Einblick hat. Es werden keine dynamische Bibliotheken, Systemvariablen benutzt oder Einträge in Register- bzw. Registrierungsdateien vorgenommen.

DAPROA ist eine Einzelplatzlösung. Es arbeitet unabhängig von lokalen oder globalen Netzwerken und Servern.

DAPROA baut von sich aus keine Verbindung zum Internet auf, nicht während der Arbeit mit ihm und auch nicht wenn es nicht genutzt wird.

Es gibt in DAPROA mit Absicht keine Links zum Internet.

DAPROA kann leicht installiert werden, indem das bereitgestellte .komprimierte Archiv in einem beliebigen Verzeichnis entpackt wird. Alle Bausteine oder Dateien, die DAPROA, braucht liegen in diesen Unterverzeichnissen. Das Entfernen geschieht ebenso einfach durch Löschen der Verzeichnisse und Dateien.

DAPROA ist bewusst konventionell aufgebaut worden. Der Anwender hat immer die Kontrolle über die Abläufe im System. Es geschieht in überschaubaren Schritten nur das, was er selbst startet. Wichtig ist, dass sich der Anwender aktiv mit den Verzeichnissen und Dateien von DAPROA auseinandersetzt.

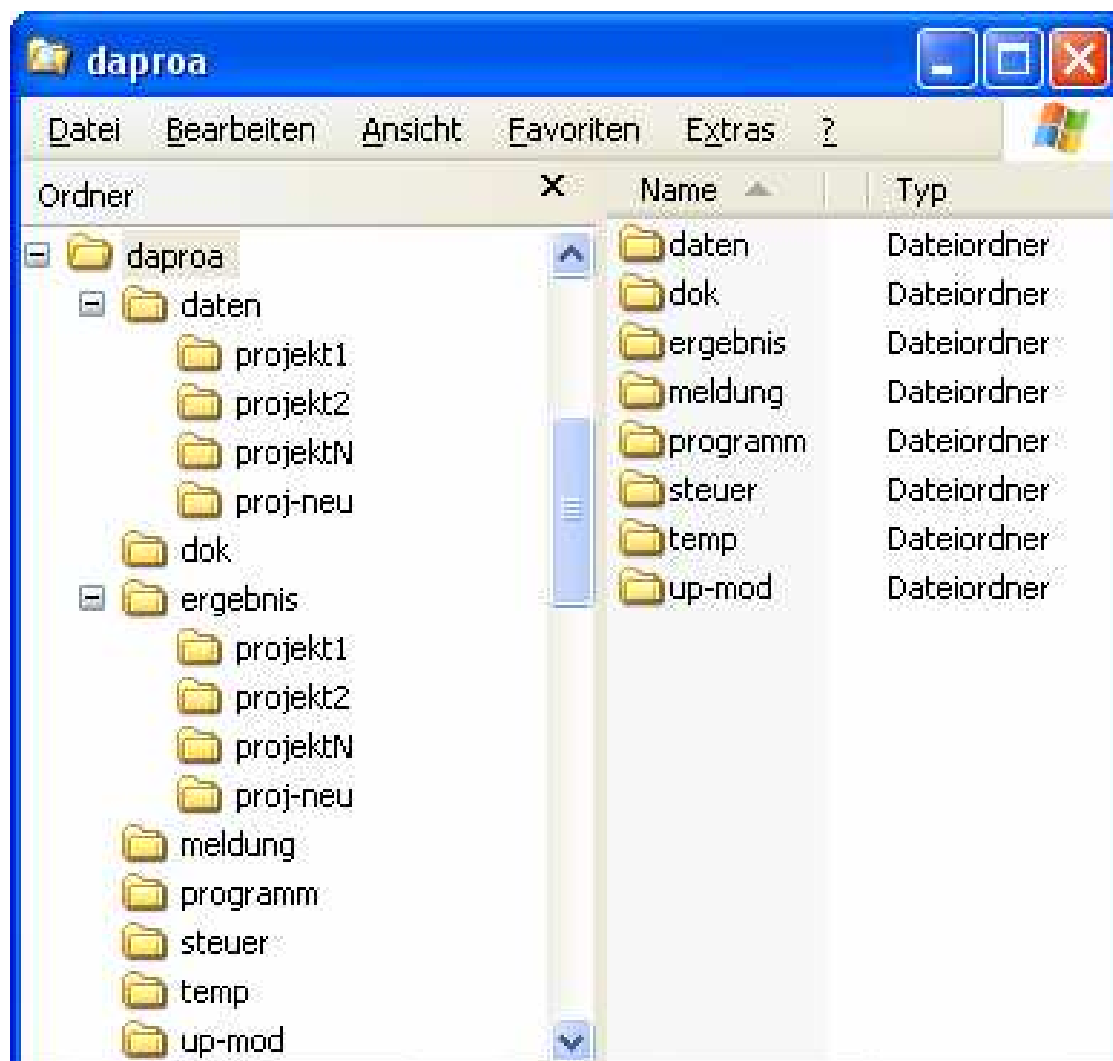
Verarbeitungsbausteine und Funktionen des Programmsystem DAPROA

Baustein	Funktion / Inhalt	zugehöriger Dialogbaustein
Ebene 0	Basis-Dialog	
PRODIA	Einstieg in DAPROA, Dialogsteuerung	
START	Aufruf der Verarbeitungsbausteine und Bereitstellung der Projekte	
Ebene 1	Behandlung Originaldateien und Dateioperationen	
NORM	Einlesen Originaldaten, Umwandeln in für DAPROA normierte Dateien. Berechnung der Mittelwerte und gegenseitigen Korrelationskoeffizienten der Variablen(Messgrößen) in der Messwertdatei bzw. des Datenbestandes.	NORM_e
TRANS	Variablentransformation: Erzeugen neuer Größen(Variablen) aus den im Datenbestand vorhandenen durch elementare mathematische Operationen und Funktionen. Löschen von Größen.	TRANS_e
SELSATZ	Selektieren(Löschen) von Datensätzen aus der Datenmatrix eines Projektes	SELSTZ_e
FKTGEN	Bildung eines Funktionstermmodells zur Verwendung in REGFKT, MODSIMA, EVOPGRD	
Ebene 2	Darstellung der Messwerte/Variablendaten	
STRDIA	Streudiagramme der Projektdaten: eine ausgewählte Größe gegen andere ausgewählte Größen jede ausgewählte Größe gegen jede ausgewählte Größe alle Größen gegen alle Größen	
GRELEM	elementare Grafik der Projektdaten, Verläufe von Variablen: eine Größe gegen eine Größe eine Größe gegen zwei Größen eine Größe gegen mehr als zwei Größen mehrere Größen in ihrer Dateireihenfolge	
GRDATNI	Niveaulinien aus ungeordneten Daten	
ZEITREI	Untersuchung und Darstellung von Zeitreihen	
Ebene 3	Parameterschätzung: lineare und quasilineare Regression	
LIREV	lineares Modell: lineare Koeffizienten, lineare Variablen	LIREV_e
POLMOD	Polynommodell: lineare Koeffizienten, polynomiale Variablen	POLMOD_e
REGFKT	Funktionstermmodell: lineare Koeffizienten, Funktionsterme	REGFKT_e

Ebene 4	Modellsimulation und grafische Darstellung	
GRBEGED	Grafik berechnet gegen gemessen	
GRKURD1	Kurvenschargrafik	
GRNIVD1	Niveauliniengrafik	
GRNIV3D	pseudo-3D Grafik einer Ergebnisfläche/ eines Flächenprofils	
Ebene 5	Modellierung und nichtlineare Parameterschätzung	
MODSIMA	Simulation von Modellen aus LIREV, POLMOD, REGFKT, FKTGEN, EVOPGRD (Modelltypen 1 – 3)	MDSMA_e
EVOPGRD	Funktionstermmodell: nichtlineare Koeffizienten, Funktionsterme, Modelltyp 3	EVPGRD_e
Ebene 6	Simulation, Modellierung und nichtlineare Parameterschätzung, Modelltyp 4	
MODSIM4	Simulation von Modellen	MDSMA_e
EVOPGRD4	explizites Modell: nichtlineare Koeffizienten, Funktionsterme	EVPGR4_e
	Hilfsprogramme	
SUCHDAT	Suchen und Anbieten der Originaleingabedatei	

Das Verzeichnissystem von DAPROA

Das Basisverzeichnis ist **daproa**



Dieses Verzeichnis kann an beliebiger Stelle auf dem Datenträger angelegt werden. Direkt in der Wurzel oder als Unterverzeichnis. In den Programmbausteinen wird mit relativen Pfaden gearbeitet.

Folgende Verzeichnisstruktur ist unterhalb von **daproa** Pflicht:

Anwenderversion:

Verzeichnis	Inhalt des Verzeichnisses
daproa	Hauptverzeichnis
daten	enthält pro Projekt ein Unterverzeichnis mit dessen Daten
projekt1	Daten und Werte des Projektes 1 Projektarbeitungsdatei des Projektes 1: projekt1_abd.txt Steuerdaten der Programme, mit denen das Projekt arbeitet.
projekt2	"
.....	
projektN	"
ergebnis	enthält pro Projekt ein Unterverzeichnis mit dessen Ergebnissen
projekt1	Protokolldateien und Ergebnisdateien des Projektes 1
projekt2	"
.....	
projektN	"
dok	Dokumentation
meldung	Protokolldateien der Module(Programme) von DAPROA: programm.mld Meldungen und Protokolldaten der Programme
programm	ausführbare Programme (Dateien Typ .exe) der Dialog- und Verarbeitungs- Programmbausteine
steuer	Steuerdateien für den Ablauf von DAPROA, seiner Module und der Projekte und Initialisierungskonstanten <ul style="list-style-type: none"> ○ prjlst_abl.txt: Namen der angelegten Projekte und Markierung des aktuellen Projektes ○ propar_ini.txt: Parameter und Konstanten von DAPROA, sie dürfen nicht verändert werden, außer Name und Pfad zu externem Dateibetrachter(Viewer) sie stehen hinter dem Wertesymbol <externer Betrachter> c:\werkzeug\wdviewer\OFFICE11\WORDVIEW.EXE, dieser Eintrag ist Standard und kann verändert werden, sollten der Dateibetrachter oder der Pfad nicht existieren ○ grafdos1d_ini.txt: Parameter und Konstanten der Grafikbausteine von DAPROA, sie dürfen nicht verändert werden ○ prgauf_inf.txt: Steuerinformationen für den Dialog ○ lizenz.txt: Lizenzinformationen für DAPROA ○ suchdat_ttt.txt: temporäre Datei für den Dialog
up-mod	Informationsverzeichnis für die Modell-Unterprogramme, welche in den Bausteinen MODSIM4 und EVPGRD4 benutzt werden
temp	Zwischenablage für temporäre Dateien, der Inhalt kann nach Ende der Sitzung gelöscht werden

Achtung !
Lege eine
Sicherungskopie
des originalen
Verzeichnisses
steuer an

Entwicklerversion: (Zur Entwicklerversion kommen noch Verzeichnisse hinzu)

Verzeichnis	Inhalt des Verzeichnisses
daproa	Hauptverzeichnis
programm 1	Quelldatei, Objektdatei und Entwicklungssteuerung
.....	
programm M	“
Libevgrd	Objektbibliotheken der Unterprogramme
Libgenp	
Libgrfd	
Libkonfi	
Liblinpa	
Libparli	
Libmod	
Libpars	
Libproa	
UP-evgrd	Quellverzeichnisse der Unterprogramme
UP-genp	
UP-grfd	
UP-konfi	
UP-linpack	
UP-mod	
UP-parlib	
UP-parsim	
UP-daproa	
UP-servm	

Die Ablaufsteuerung in DAPROA

Das Projekt, Projektdateien

Einem Projekt wird ein maximal 8-stelliger Kurzname zugewiesen. Dieser Kurzname dient als Name für die Verzeichnisse und Projektdateien, sowie Steuerdaten und wird nachfolgend in diesem Dokument mit Projektname bezeichnet.

Die Variablennamen, Projektnamen und auch andere selbstbenannte Modell- und Dateinamen müssen aus dem Zeichenvorrat des ASCII-Zeichen-Codes (Dezimal) von 48-57 (Ziffern), 65-90 (Großbuchstaben) und 97-122 (Kleinbuchstaben) sowie 45 (Bindestrich) gebildet werden. Umlaute und ß sind nicht erlaubt.

Bei Nichtbeachtung dieser Forderung kommt es zu schweren Störungen bei der Dateiarbeit!

Insbesondere die Verwendung der Zeichen 95 , 46 , 44 , 59 , 43 , 42 , 32 (_ Unterstrich, . Punkt, , Komma, ; Semikolon, + plus, * mal, leer sowie Querstriche, Klammern und andere Sonderzeichen) führt zu Fehlern, da sie u.a. programmintern als Positionszeichen zum Aufbau der Dateinamen dienen oder auch in Steuerelementen benutzt werden.

Die Arbeit mit einem Projekt steht im Mittelpunkt des Ablaufes von DAPROA. Es kann immer nur ein Projekt aktuell bearbeitet werden. Das aktuelle Projekt wird beendet, wenn während einer DAPROA-Sitzung ein anderes Projekt gestartet werden soll. Die Steuerdaten des beendeten Projektes bleiben gespeichert. Im Dialogbetrieb geschieht dies automatisch.

Projektarbeit im Dialog :

Der Name des originalen Projektes (Haupt-Projekt, welches zum Projektstart angelegt wurde) und der Name der Unterverzeichnisse, in denen die Daten und Ergebnisse des Projektes stehen müssen gleich sein!

In den jeweiligen Projektverzeichnissen befinden sich nach Bedarf aber auch Daten und Ergebnisse von sog. Unterprojekten oder Modellen, welche einen anderen Namen tragen. Das originale Projekt(Hauptprojekt) ist die Klammer über die Unterprojekte, die aus ihm heraus entstanden sind.

Ein Unterprojekt, hat wie das Hauptprojekt, nach ihm benannte Projektbasisdaten. Das originale Projekt wird während der Verarbeitung selbst wie ein Unterprojekt behandelt.

Der Dialog

Der Dialog wird über den Programmbaustein PRODIA gestartet.

PRODIA ruft den Programmbaustein START auf, mit welchem der Zugriff auf die Verarbeitungs- Programmbausteine gesteuert wird.

Innerhalb dieses Dialoges wird vor jedem gerufenen Programmbaustein ein zugehöriger Dialogbaustein gestartet.

Im Gegensatz zu den Verarbeitungs- Programmbausteinen haben die reinen Dialogbausteine außerhalb des Dialoges keine Wirkung.

Die Kommunikation zwischen PRODIA, START, den Verarbeitungs- Programmbausteinen und ihren zugehörigen Dialogbausteinen geschieht nur über Dateien, die abwechselnd geschrieben und gelesen werden.

Diese Dateien stehen im Verzeichnis `daproa\steuer` :

- `prgauf_inf.txt`: Steuerinformationen für den Dialog
- `suchdat_ttt.txt`: temporäre Datei für den Dialog

Ablauf einer Projektbearbeitung

Der Ablauf einer Projektbearbeitung in einer DAPROA-Sitzung ist wie folgt kurz unter (1), (2), (3) beschrieben. Alle diese nachstehenden Aktivitäten können im Dialog durchgeführt werden bzw. werden im Dialog automatisch vorgenommen. Es wird vorausgesetzt, dass die für den Baustein notwendigen Eingabedateien vorhanden sind.

(1) neues Projekt eröffnen:

- Festlegen eines max. 8-stelligen Projektnamens durch den Nutzer im Baustein START
- Anlegen des Projektverzeichnisses `..\daproa\daten\projektname`
- Anlegen des Projektverzeichnisses `..\daproa\ergebnis\projektname`

- Eingetragen des Projektnamens in die DAPROA-Ablaufdatei `prjlst_abl.txt`. Diese Datei steht im Verzeichnis `.\daproa\steuer`
- Bereitstellen der Datei mit den originären Ursprungsdaten und daraus gebildeter Dateien entsprechend der Vorschrift zu den Ein- und Ausgabedateien des Programmbausteins NORM
Die Datei mit den originären Ursprungsdaten kann an beliebiger Stelle des Datenträgers stehen, aber natürlich auch im Projektverzeichnis `..\daproa\daten\projektname`

(2) neues Projekt zum ersten Mal abarbeiten:

- Anlegen der Projektarbeitungsdatei `projekt_abd.txt` in `..\daproa\daten\projektname`
- In diese Projektarbeitungsdatei die Steuerdaten für den Programmbaustein NORM eintragen
- In der DAPROA-Ablaufdatei `prjlst_abl.txt` ist das Projekt als aktuell zu kennzeichnen.
In diesem Fall steht in der ersten Spalte der Zeile mit dem Projektnamen das Zeichen '<' .
Die nicht aktuellen Projektnamenszeilen müssen in der ersten Spalte das Zeichen '*' haben.
- Nach einem erfolgreichen Aufruf des Programmbausteins NORM entstehen im Verzeichnis `..\daproa\daten\projektname` die Dateien
 - `projekt_abd.txt` Projektarbeitungsdatei des Projektes
 - `projekt_mtx.txt` Matrixdatei der Messwerte/ Untersuchungsdaten
 - `projekt_nqd.txt` Normdaten Eingabedatei
 - `projekt_std.txt` Steuerdatei BasisstatistikSie sind Voraussetzung für die Arbeit der anderen Bausteinen.
Diese Dateien müssen nicht mit dem Programmbaustein NORM erzeugt werden.
Man kann sie können auch mit anderen, selbst gewählten Mitteln bereitstellen.
Allerdings ist NORM genau für diesen Zweck geschaffen worden.

(3) ein "altes" Projekt mit einem Programmbaustein abarbeiten:

- In der DAPROA-Ablaufdatei `prjlst_abl.txt` ist das Projekt als aktuell zu kennzeichnen.
In diesem Fall steht in der ersten Spalte der Zeile mit dem Projektnamen das Zeichen '<' .
Die nicht aktuellen Projektnamenszeilen müssen in der ersten Spalte das Zeichen '*' haben.
- In die Projektarbeitungsdatei `projekt_abd.txt` in `..\daproa\daten\projektname` die Steuerparameter des Programmbausteins eintragen.

Ein Projekt löschen

Aus der Liste der originalen Projekte, die in der Datei `prjlst_abl.txt` steht, werden nicht mehr benötigte Projekte **nicht** automatisch gelöscht. Wenn also der Fall eintritt, dass die Dateien und/oder Projektunterverzeichnisse eines solchen Projektes in den Verzeichnissen `\daten` und `\ergebnis` gelöscht werden, muss der Projekteintrag auch aus der Datei `prjlst_abl.txt` durch Löschen der Zeile entfernt werden.

Der Dialog für den Ablauf von DAPROA



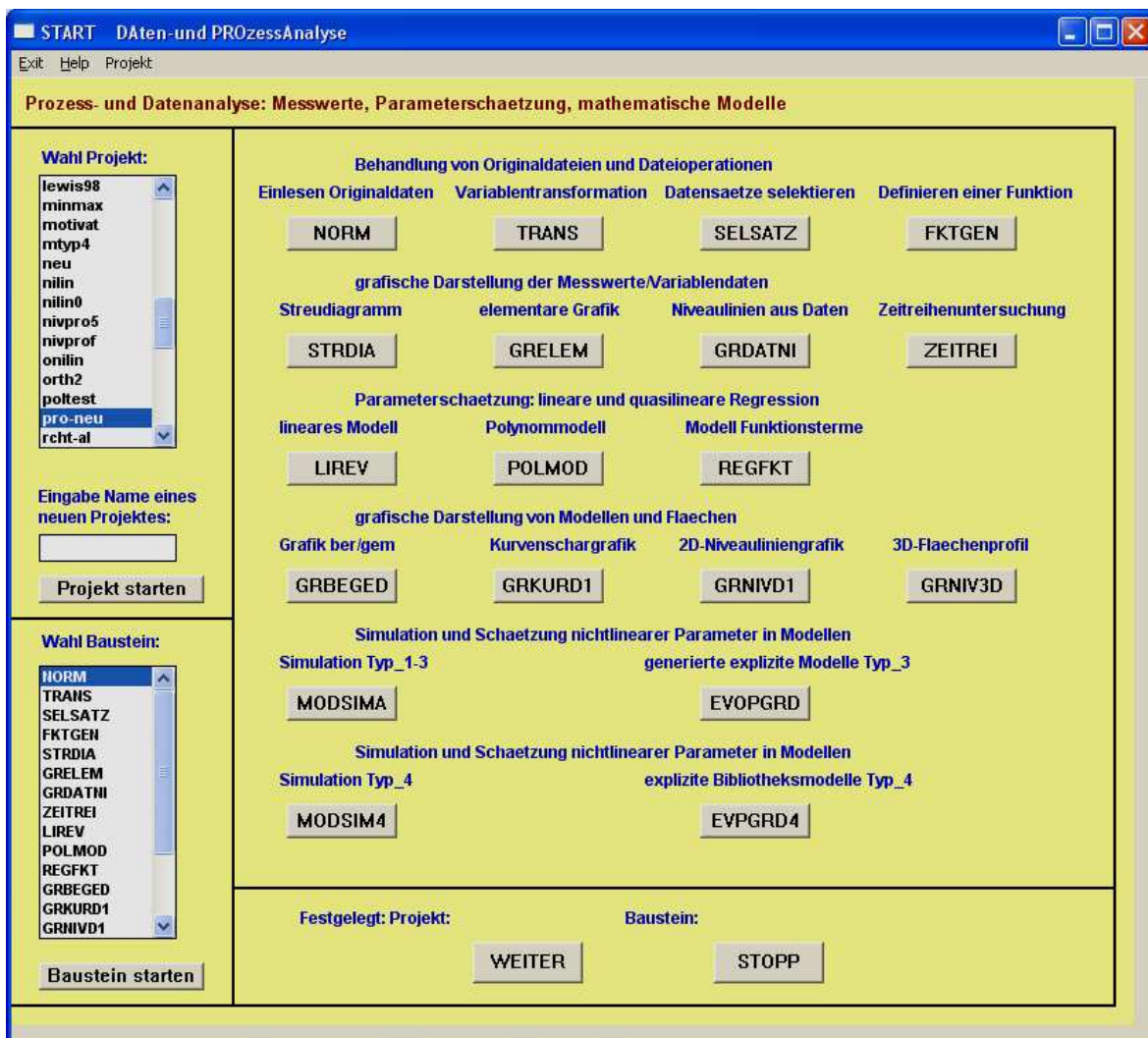
Um das Programmsystem DAPROA zu eröffnen wird der Programmbaustein PRODIA gestartet.

Eine Verknüpfung zu PRODIA mit einer Ikone bzw. einem Symbol auf dem Desktop



dargestellt erleichtert den Start.

Nachstehend ist der Startbildschirm von DAPROA dargestellt. Zur Bedienung siehe in der Dokumentation zum Baustein START.



In PRODIA wird zuerst der Programmbaustein START als paralleler Prozess gerufen. Von START aus können die anderen Dialogbausteine ausgewählt werden, welche wiederum ihren zugehörigen Verarbeitungs- Programmbaustein starten. Nach erfolgreicher Arbeit des Verarbeitungs- Programmbausteins kehrt der Dialog wieder zu seinem Dialogbaustein zurück.

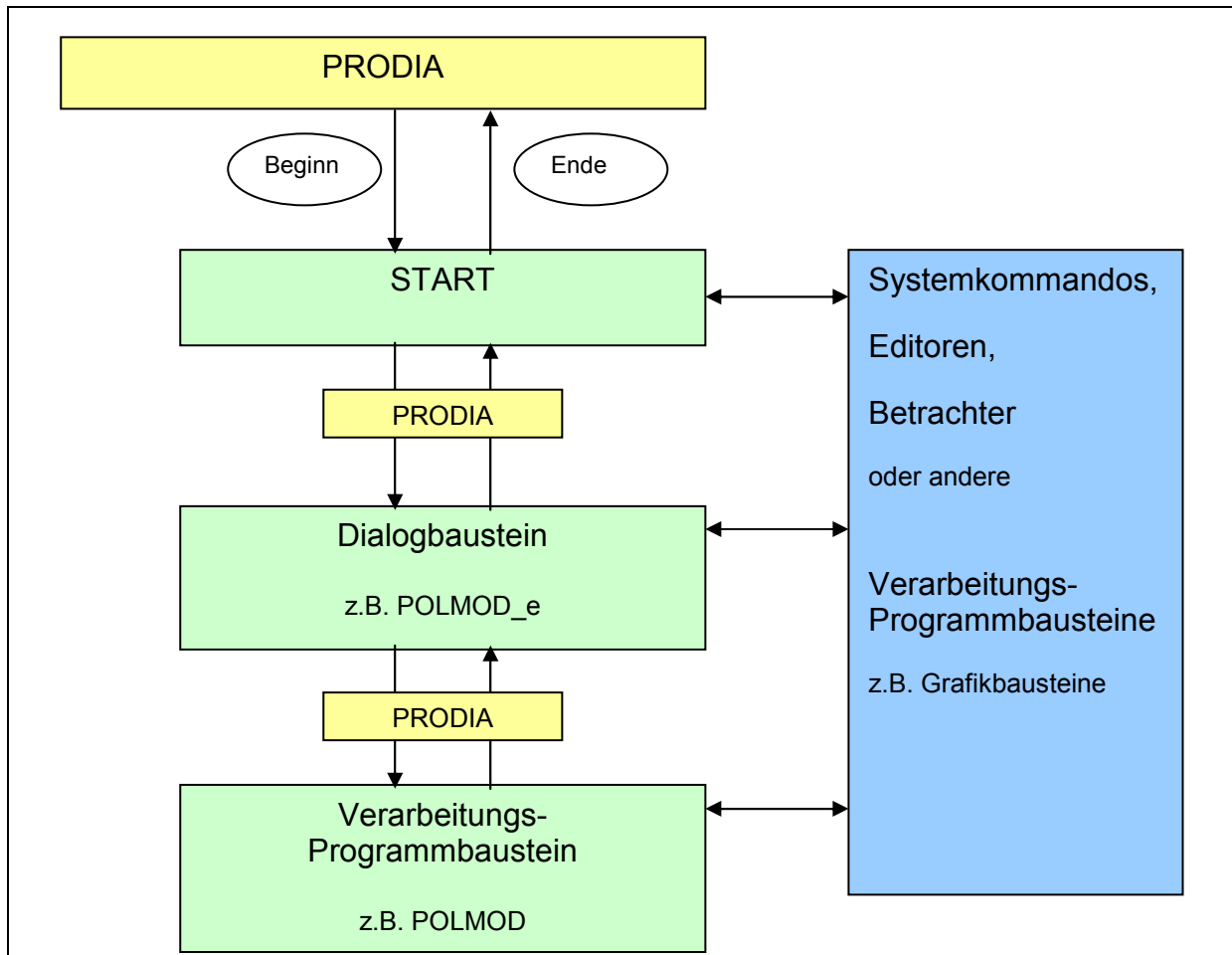
Von den Dialog- oder Verarbeitungs- Programmbausteinen aus werden auch weitere parallele Prozesse gestartet, so für Systemkommandos, Editoren, Betrachter oder andere Verarbeitungs- Programmbausteine. Diese parallelen Prozesse werden sind bewusste und fest verankerte Bestandteile der Verarbeitungs- Programmbausteine. Nach der Beendigung eines parallelen Prozesse kehrt die Steuerung wieder zum rufenden Baustein zurück.

Es können also mehrere Prozesse gleichzeitig aktiv sein

- PRODIA
- START oder Dialog- oder Verarbeitungs- Programmbausteine
- Systemkommandos, Editoren, Betrachter oder andere Verarbeitungs- Programmbausteine entspr. der Wahl durch den Baustein

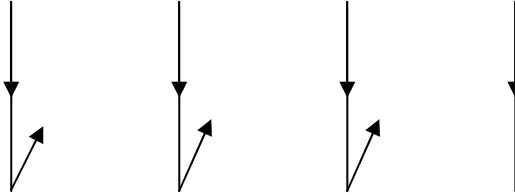
Der Aufruf der parallelen Prozesse kann zu Fehlern führen, wenn nicht genügend Systemspeicher frei ist.

Schematischer Ablauf in UPROD2A



prinzipieller Ablauf der Arbeit mit einem Dialogbaustein

Ein Dialogbildschirm ist in mehrere Abschnitte unterteilt.
Der logische Ablauf geht in der Regel von oben nach unten (erstrangig)
und dann von links nach rechts (zweitrangig)



Die einzelnen Dialogblöcke sind durch Rahmen voneinander getrennt.

Wesentliche Eingaben müssen per Taste/Button bestätigt werden.
Im untersten Teil des Bildschirms befinden sich in der Regel die Tasten/Buttons für die Dialogsteuerung.

Die im Dialog erfassten Daten und Steuerparameter werden gespeichert, wenn die Taste **WEITER** oder eine andere spezielle Speichertaste bzw. Starttaste betätigt wird.

Nach der erfolgreichen Prüfung werden die Daten in die Projektarbeitsdatei **projekt_abd.txt** mit dem entspr. Kennzeichen des zugehörigen Programmbaustein geschrieben.

Der letzte Inhalt der Projektarbeitsdatei **projekt_abd.txt** kann in der Regel in das Menü des Dialogbausteins geladen werden.
Die entsprechende Taste/Button befindet sich dann meist ganz oben im Menü.

Die obere Menüleiste

Im obersten Maskenbereich ist eine Menüleiste angeordnet. Die Menüfunktionen können unabhängig vom Fortschritt des Dialoges angesprochen werden.

In der Regel angebotene Menüpunkte im Beispiel:

Menüpunkt	Funktion	Inhalt
Exit	OK	Ende des Programms, es wird aber empfohlen die Stopp_Taste in der Dialogmaske zu benutzen, da sonst das Dialogsystem eine Fehlermeldung ausgibt.
	Quit	Ende des Programms, es wird aber empfohlen die Stopp_Taste in der Dialogmaske zu benutzen.
Help		Aufblenden einer kurzen Hilfeanleitung zur Benutzung der Dialogmaske, sie ersetzt nicht die Dokumentation.
Ergebnis, Datei oder andere	Modell, Protokoll, Projektbasis- daten oder andere	Anzeigen der Textdatei welche die entsprechenden Daten des aktuellen Objektes enthält. Das Anzeigen geschieht in einem separaten Fenster. Es werden maximal 10000 Zeichen oder 65 Zeilen auf- geblendet. Zu große Dateien werden abgeschnitten. Die Namen vom Modell, Protokoll, .. müssen eventuell auf der Dialogmaske angegeben oder gewählt worden sein.

Konstanten, Parameter

<Auswahl / Informationen aus propar_ini.txt>

Konstante/ Parameter	Erläuterung	Wert Vollversion	abweich. Wert Leichtversion
mstz	max.Anzahl von Datensätzen : allgemein	9.999.999	20
	max.Anzahl von Datensätzen: NORM, GRDATNI, ZEITREI, TRANS	9.999.999	100
mvar	max.Anzahl der Variablen	100	3
itmax	max. Anzahl von Projektsteuersätzen (allg.in _abd) max. Anzahl von Transformationen (TRANS)	100	
	max. Anzahl von Projektsteuersätzen (allg.in _abd) max. Anzahl von Transformationen (TRANS)	99	
mprgr	max.Anzahl von Projekten / Modellen	100	
mnam	max.Länge des Variablennamens	15	
mwrt	max.Länge des Werteformates	15	
mjew	max.Anzahl Sätze für gültige Eingabewerte	10	
zfehl	Ersatz für fehlenden Wert	-1.D50	
ugrenz	absoluter unterer Grenzwert für Messwert/ Untersuchungswert	1.D-30	
ogrenz	absoluter oberer Grenzwert für Messwert/ Untersuchungswert	1.D30	
epsm	Grenzwert Multiplikationsparameter Epsilon	1.D-10	
epsl	Grenzwert logarithmischer Parameter	1.D-05	
epse	Grenzwert Exponentialparameter	1.D-03	
mxtm	max.Anzahl von Termen im Regressionspolynom	99	10
mxgr	max.Grad eines Terms im Regressionspolynom	5	2
mxeg	max.Anzahl von Einflussgrößen im Regressionspolynom	12	2
mfak	max.Anzahl Faktoren im Term eines Regressionspolynoms	5	2
mmg	max.Anzahl von linearen Regressionskoeffizienten im Endansatz	51	6
maxstx	max. Anzahl Variationsschritte der Abszisse	100	
minstx	min. Anzahl Variationsschritte der Abszisse	20	
maxstys	max. Anzahl Variationsschritte der Kurvenscharvariablen und Niveaulinien	20	
minstys	min. Anzahl Variationsschritte der Kurvenscharvariablen und Niveaulinien	0	
maxstyn	max. Anzahl Variationsschritte der Ordinate (Niveaulinien)	100	
minstyn	min. Anzahl Variationsschritte der Ordinate (Niveaulinien)	20	
mftk	max. Anzahl von Funktionsausdrücken	10	2
mvaru	max. Anzahl von unabhängigen Variablen in Funktionsausdrücken	12	2
npog	max. Anzahl nichtlinearer Schätzparameter	20	3
maxnp0	max. Anzahl gleichzeitig schätzbarer nichtlin. Parameter	10	2
itrmx	max. Anzahl von Iterationen bei nichtlin Schätzung	1000	100
maxips	max. Anzahl Schritte bei Startschätzung	50	
jvmax	max. Verschiebung in Zeitreihen	500	50

mvar	max.Anzahl der Variablen in Simulation Typ 4	60	
mswe	max. Anzahl Messwerte GRDATNI Variante 3 und 4	5000	100
ntog	nichtlin.Modell:Obergrenze Anzahl der Einflussgrößen	20	3

Abbruch von Dialog- und Verarbeitungsbausteinen

Wenn die Programme mit einem "fatalen Error" abbrechen gibt es keine Fehlermeldung. Die Steuerung kehrt zum Baustein START, bzw. bricht auch START ab, zurück.

Ursachen können sein:

- fehlerhafte Anwendung von Funktionen und Modellen, Nichtentsprechung der zu verarbeitenden Daten bezogen auf die eingesetzten Funktionen und Modelle, Verletzung der Definitionsbereiche der benutzten mathematischen Ausdrücke:
Beispiele: Überlauf in der Exponentialfunktion
Division durch Null
Quadratwurzel bzw. Logarithmus aus negativer Größe
- nicht reguläre Daten oder Verzeichnisse . Sie verursachen evtl. unerlaubte Operationen
- eine fehlerhaft veränderte Projektarbeitungsdatei **projekt_abd.txt**
- fehlerhaft veränderte **_ini.txt** Steuerdateien
- Programmfehler
- Speicherbereichsüberlauf bzw. -verletzung

DAPROA führt eine Vielzahl von logischen, Grenzeinhaltungs-, Datenformats- und andere Prüfungen durch.

In einer Dialogbox wird vor negativen Prüfungsergebnissen gewarnt. Bei schwerwiegenden Verletzungen erfolgt ein geordneter Abbruch der aktuellen Verarbeitung. Trotz aller Sorgfalt können Fälle auftreten, deren Daten- und Parameterkonstellation im Test nicht durchlaufen wurden und zu einem unkontrollierten Abbruch führen. DAPROA bittet, solche Fälle mit Meldungsdatei **baustein_mld.txt** , Projektarbeitungsdatei **projekt_abd.txt** , und Datendateien **projekt.txt** , **projekt_std.txt** , **projekt_mtx.txt** (eventuell die Namen anonymisiert) zur Prüfung und Fehlersuche zu senden.

Die Historie von DAPROA

Dieses Vorhaben auf standardisierte Dateien, mit Versuchs-, Prozess- oder sonstig gesammelten Daten als Inhalt, verschiedene Bausteine (mathematische und grafische Methoden) zur mathematischen Analyse und Auswertung im Zusammenspiel anzuwenden ist schon mit dem Vorhandensein leistungsfähiger Rechner in den 1960er zu 70er Jahre entstanden.

- Die erste Realisierung eines Vorläufersystems von DAPROA erfolgte auf Großrechnern mit endlosen Ergebnislisten und bescheidenen Grafiken. Der Anwender-Dialog wurde mittels Jobsteuerung im Batch, dann auch per Bildschirm ausgeführt.

- Eine zweite Realisierung geschah auf nun leistungsfähigen Personalcomputer ab Mitte der 1980er Jahre unter DOS mit einer Dialogführung auf zeichenorientierten Bildschirmen, aber schon recht ordentlichen grafischen Auswertungen, entsprechend den damaligen rechentechnischen Leistungsparametern.
- In der dritten Etappe sind die Bausteine ab Anfang/Mitte der 1990er Jahre für leistungsfähige 32-bit-Prozessor-Rechner und dazugehörigen Grafikmöglichkeiten entstanden.
Die Lösungen all dieser 3 Etappen wurden von Spezialisten, die in gemeinsamen Arbeitsbereichen oder in einem losen Netzwerk miteinander arbeiteten, für die Lösung von entsprechenden Aufgaben aus Forschung, Entwicklung, Technik, Prozessführung, Überwachung/Kontrolle usw. eingesetzt.
- In Fortsetzung der 3. Etappe entstand DAPROA Version 5.x als die Einheit von Dialog- und Verarbeitungsbausteinen. Damit gibt es nun die Möglichkeit DAPROA, mit seinen Leistungen und als Programmsystem, einer Allgemeinheit anzubieten. Ein Verständnis für die mathematisch gestützte Analyse und Auswertung von Daten, auch der mathematischen Modellbildung, ist aber Voraussetzung für einen sinnvollen Einsatz.

Die eingesetzten mathematischen Verfahren haben sich im Laufe der Jahre weiterentwickelt, sind eine Fortsetzung derer, die anfangs verwendet wurden. Die Rechengeschwindigkeit und die Handhabung der Bausteine haben sich natürlich wesentlich verbessert. Alle Methoden und Verfahren wurden bei jeder Etappe neu algorithmiert und die Bausteine auch seitens ihrer Programmquelltexte immer wieder neu aufgebaut, so auch die Dialog- und Verarbeitungssteuerung. Die Idee (das Konzept) ist es, welches über einen langen Zeitraum weitergeführt wurde.