

# **UMDBS**

**Universal Micro DataBase System**

## **Benutzerhandbuch**

**Technische Universität Darmstadt  
Fachgebiet Statistik und Ökonometrie  
Univers.-Prof. Dr. Hans-Dieter Heike**

**Prof. Dr. Thomas Sauerbier**

**Version 3.0**

**3.9.2001**

# Inhaltsverzeichnis

<b>1</b>	<b>EINFÜHRUNG .....</b>	<b>4</b>
<b>2</b>	<b>SYSTEMÜBERBLICK.....</b>	<b>5</b>
2.1	Funktionsbereiche .....	5
2.2	Benutzeroberfläche und Menüs.....	5
2.3	Dateien .....	6
2.4	Online-Dokumentation.....	8
<b>3</b>	<b>FLUBDIAGRAMME.....</b>	<b>9</b>
3.1	Allgemeines.....	9
3.2	Module .....	10
3.2.1	Mikrodatenbasis .....	10
3.2.2	Simulationsmodell.....	10
3.2.3	Parameterdatenbasis .....	11
3.2.4	Zeitreihendatenbasis .....	11
3.2.5	Report .....	11
3.2.6	Verteilungsdatenbasis.....	11
3.2.7	Analysemodell .....	12
3.2.8	Testmodell.....	12
3.2.9	Generierungsmodell .....	12
3.3	Simulation .....	13
3.3.1	Standard-Simulation.....	13
3.3.2	Erweiterte Simulation.....	14
3.3.3	Monte-Carlo-Simulation .....	15
3.4	Analyse.....	16
3.5	Generierung / Import.....	17
<b>4</b>	<b>EDITOREN.....</b>	<b>19</b>
4.1	Textbasierende Editoren.....	19
4.2	Mikroobjekt-Editor .....	21
<b>5</b>	<b>MONITORE .....</b>	<b>23</b>
5.1	Mikroobjekt-Monitor .....	23
5.2	Mikro-Monitor .....	24
5.3	Verteilungs-Monitor.....	25
5.4	Zeitreihen-Monitor .....	26
5.5	Makro-Monitor.....	26
5.6	Monte-Carlo-Monitor.....	27
5.7	Parameter-Monitor .....	28
<b>6</b>	<b>HILFSPROGRAMME .....</b>	<b>30</b>
6.1	SOEP-Dictionary-Monitor .....	30
6.2	Datei-Monitor.....	31
6.3	Import-Editor.....	32

<b>ANHANG .....</b>	<b>34</b>
A.1 Installation.....	35
A.2 Lizenzrechtliche Hinweise .....	36
A.3 Grenzen der aktuellen Implementierung .....	37
A.4 Änderungen zu früheren Versionen des UMDBS.....	38
<b>INDEX .....</b>	<b>39</b>

## 1 Einführung

Mit dem UMDBS steht ein System zur Arbeit mit querschnittsorientierten Mikrodatenbeständen zur Verfügung, das die Bereiche Import/Generierung, Analyse und Simulation abdeckt. Gegenüber den meisten anderen Mikrosimulationssystemen sind vor allem folgende Vorteile zu nennen:

- Das UMDBS ist ein System, das ohne zusätzliche Software und ohne lizenzrechtliche Probleme auf jedem heutigen PC (unter Windows 95, 98, NT und 2000) lauffähig ist. Demgegenüber setzen die meisten anderen Systeme entweder Großrechner oder UNIX-Workstations voraus oder laufen nur in Verbindung mit einer anderen, meist sehr teuren Software (z.B. Datenbanken oder Statistikpaketen), die der Anwender selbst getrennt erwerben müßte.
- Seine Flexibilität und Leistungsfähigkeit verdankt das UMDBS vor allem der darin integrierten Mikromodellierungssprache MISTRAL (Micro Simulation, Transformation, and Analysis Language), für die ein Compiler und eine virtuelle Maschine mit Laufzeitumgebung vorhanden sind. Die Sprache unterstützt in durchgängiger Form alle Aktionen von der einfachen interaktiven Analyse über den Import von Mikrodaten anderer Systeme bis hin zu komplexen Simulationen mit integrierten Analyse- und Test-Routinen. Für die Benutzung des UMDBS ist deshalb die zumindest grundlegende Kenntnis von MISTRAL unverzichtbar. Dafür steht ein umfassendes Handbuch zur Verfügung. Da sich die Sprache stark an das verbreitete PASCAL anlehnt, ist zudem die Einarbeitung bei Vorliegen grundlegender Programmierkenntnisse schnell und einfach möglich.
- Viele Simulationssysteme besitzen nur eine spartanische Benutzeroberfläche und lassen sich z.T. fast nur von den Entwicklern selbst sicher bedienen. Das UMDBS basiert hingegen einerseits auf Windows-Standards. Andererseits erleichtern weitgehend intuitiv bedienbare Flußdiagramme alle wichtigen Abläufen sowohl für mit der Mikrosimulation vertraute Anwender als auch völlige Anfänger auf diesem Gebiet.

Der letzte Punkt sollte es ermöglichen, daß Anwender auch ohne umfangreiches Handbuch in der Lage sind, das System sicher zu bedienen. Das vorliegende Handbuch soll deshalb vor allem einen Überblick über das System und seine wesentlichen Bestandteile verschaffen sowie die Funktionalitäten beschreiben, die nicht offensichtlich sind. Bezüglich Bedienungsdetails, die in Windows üblich sind oder sich weitgehend von selbst erklären, wurde auf eine Beschreibung verzichtet. Der Grund besteht in den beschränkten Ressourcen, die an solchen Punkten zwangsläufig Abstriche erfordern, und in der permanenten Fortentwicklung des UMDBS, die entsprechend eine ständige Aktualisierung der Dokumentation erfordern würde. Zudem steht unter dem Menüpunkt "Hilfe" inzwischen ein Übungshandbuch (Tutorial) zur Verfügung, das eine rasche Einarbeitung anhand praktischer Beispiele ermöglicht.

## 2 Systemüberblick

### 2.1 Funktionsbereiche

Das UMDBS unterstützt insbesondere das Erzeugen (inkl. Import), Analysieren und Simulieren von (querschnittsorientierten) Mikrodaten. Sofern für diese Funktionen mehrere Arbeitsschritte bzw. Dateien erforderlich sind, werden diese Abläufe in einem Flußdiagramm zusammengefaßt, von dem aus besonders einfach alle benötigten Werkzeuge aus aufgerufen werden können.

Zusätzlich gibt es drei Gruppen von Werkzeugen, die auch außerhalb der Flußdiagramme allein verwendet werden können:

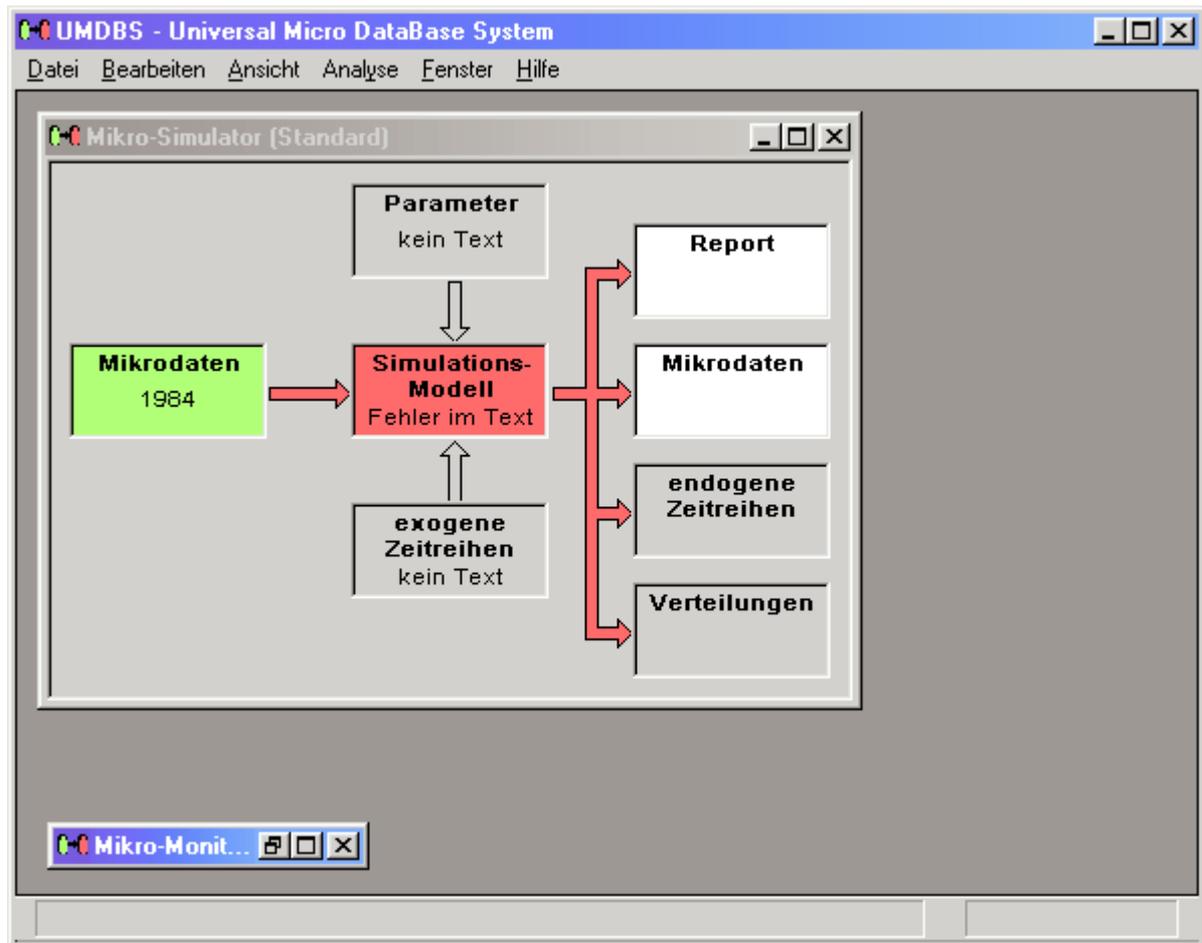
Editoren erlauben das Eingeben und Verändern von Programmen und Datenbeständen. Aktuell sind spezialisierte Texteditoren für die Bearbeitung von MISTRAL-Programmcode sowie ein Editor für die direkte manuelle Bearbeitung von Mikrodaten verfügbar.

Monitore erlauben eine interaktive Analyse verschiedenartiger Datenbestände, z.B. von Mikrodaten, Zeitreihendaten, Fortschreibungsparametern und unterschiedlichen Simulationsergebnissen. Da die Ergebnisse einer Simulation abgespeichert werden können, lassen sich die Monitore für Simulationsergebnisse nicht nur direkt im Anschluß an eine Simulation, sondern anhand der erzeugten Dateien auch später nutzen.

Zusatzprogramme dienen der Erweiterung der Möglichkeiten und des Komforts des UMDBS. Aktuell ist ein Hilfsprogramm zur Analyse von SOEP-Dateien (mit automatischer Programmgenerierung für den Import) sowie ein Programm zum Import von DPMS-Parametern verfügbar.

### 2.2 Benutzeroberfläche und Menüs

Das UMDBS präsentiert sich in Form einer MDI-Oberfläche, wie sie von vielen Windows-Programmen her bekannt ist. Dabei können innerhalb eines Gesamtfensters viele Teilfenster parallel geöffnet oder als Symbole verkleinert dargestellt werden.



Der Aufruf der Programmteile erfolgt unterhalb des Menüs "Datei | Werkzeuge". Sind Werkzeuge aktiv, so können in vielen Fällen dazugehörige Programmteile von dort aus automatisch geöffnet werden.

Alle wichtigen Funktionen des aktuell aktiven Fensters können jeweils über die Menüs am oberen Rand des Gesamtfensters aufgerufen werden. Zusätzlich macht das UMDBS intensiv von kontextsensitiven Popup-Menüs (über rechte Maustaste) Gebrauch.

## 2.3 Dateien

Alle Dateien, die vom UMDBS eingelesen oder erzeugt werden, beinhalten Daten im ASCII-Format. Dabei handelt es sich teilweise um Text, der von Benutzern erzeugt oder gelesen werden soll (z.B. MISTRAL-Programme oder Reports), teilweise auch um Dateien, die nur vom System erzeugt und gelesen werden. Für Zwecke des Imports oder Exports sowie für spezielle Zwecke (z.B. Debugging) können auch diese Dateien manuell oder von anderen Programmen bearbeitet werden. Es ist jedoch zu beachten, daß anhand des Dateikopfes (erste ein bis zwei Zeilen) geprüft wird, ob es sich um eine Datei des erwarteten Typs handelt. Wenn ja, wird vorausgesetzt, daß der Rest der Datei exakt der Spezifikation entspricht. Im Falle von Abweichungen, die durch manuelle Eingriffe entstanden sind, kann das Programm ebenso unvorhersehbare Reaktionen zeigen, wie Programme, die korrupte Binärdateien einlesen.

Innerhalb des Systems werden folgende Dateitypen verwendet:

- \*.dis Verteilungsdatenbasis (DIStribution)  
Wird ausschließlich vom System erzeugt und gelesen.
- \*.mcd Ergebnisdatei einer Monte-Carlo-Simulation (Monte Carlo Database)  
Wird ausschließlich vom System erzeugt und gelesen.
- \*.mml Datei mit MISTRAL-Programm (Micro Modelling Language)  
Vom Benutzer erzeugte Programmdateien für Simulation, Analyse, Test und Generierung/Import.
- \*.odb Mikrodatenbasis (Object DataBase)  
Wird vom System erzeugt und gelesen; Struktur jedoch im MISTRAL-Handbuch beschrieben
- \*.pdb Parameterdatenbasis  
Wird vom Benutzer erzeugt.
- \*.rpt Default-Name einer vom System erzeugten Reportdatei  
Wird vom System erzeugt (inkl. benutzerdefinierter Ausgaben eines MISTRAL-Programms) und ist ausschließlich für das Lesen durch einen Benutzer vorgesehen.
- \*.ts Zeitreihendatenbasis (Time Series)  
Wird manuell vom Benutzer oder automatisch vom System aus dem Ergebnis einer Simulation erzeugt.

Zusätzlich zu diesen UMDBS-Dateien greift das System noch auf folgende Dateitypen des Sozioökonomischen Panels (SOEP) zu:

- \*.dat SOEP-Daten-File  
Enthält die Daten der Mikroobjekte.
- \*.dic SOEP-Dictionary-File  
Enthält eine Definition der Datenstruktur (z.B. Position und Kodierung der Attribute) im dazugehörigen Daten-File.

Intern verwendet das UMDBS zwei \*.ini-Dateien:

- reginfo.ini Diese Lizenzdatei wird registrierten Benutzern vom Programmautor zur Verfügung gestellt. Ohne Diese Datei ist die Simulation auf Mikrodatenbasen mit maximal 5.000 Mikroobjekten beschränkt. Die Datei ist mit einem Schlüssel versehen und kann nicht manuell editiert werden.
- umdb.ini In dieser Datei werden allgemeine Benutzereinstellungen (unter "Datei" | "allgemeine Optionen") dauerhaft gespeichert. Zur Zeit sind dies die Sprache (Deutsch oder Englisch) und der Default-Pfad für die Benutzerdateien. Existiert die Datei nicht, wird beim Programmaufruf nach der gewünschten Sprache gefragt und die Datei damit neu angelegt.

### 2.4 Online-Dokumentation

Unter dem Menüpunkt "Hilfe" stehen verschiedene Dokumentationen und Informationen zur Verfügung:

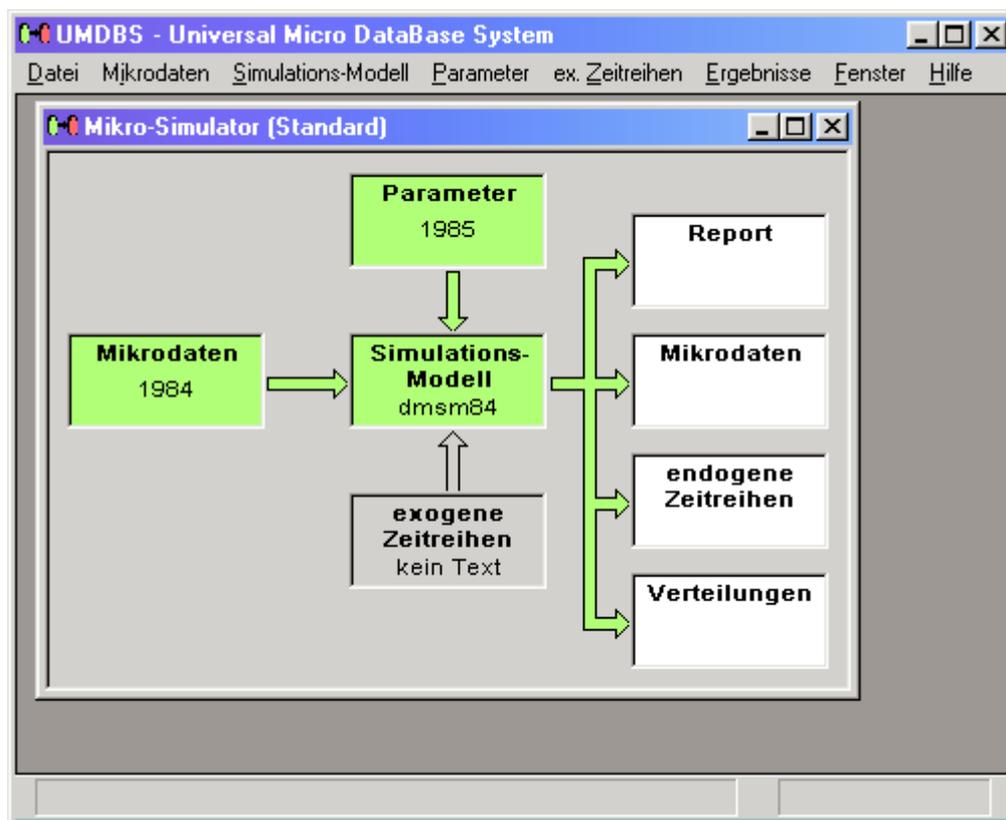
- In der Readme-Datei (readme.txt) befinden sich Informationen zur aktuellen Programmversion. Dort sind insbesondere die Änderungen und beseitigten Fehler gegenüber der Vorgängerversion aufgelistet.
- Das UMDBS-Handbuch (das vorliegende) sowie das MISTRAL-Handbuch können als PDF-Dateien direkt aufgerufen werden (in der Regel mit dem auf dem Rechner installierten Acrobat Reader).
- Ein Übungs-Handbuch ist als Windows-Hilfe-Datei verfügbar. Es enthält eine strukturierte Anleitung mit konkreten Übungsaufgaben zum Selbstlernen. Die dazu benötigten Dateien werden mit dem System ausgeliefert.
- Eine Kurz-Präsentation des Systems ist in HTML verfügbar und kann über den Default-Browser betrachtet werden.
- Ein Hinweis auf die WWW-Seite, von der die jeweils aktuellste Version des UMDBS geladen werden kann.
- Eine Kurzinformation zur Programmversion (Versionsnummer und Erstellungsdatum) sowie dem Status der Lizenzierung.

## 3 Flußdiagramme

### 3.1 Allgemeines

Flußdiagramme sind die zentralen Instrumente für den Benutzer, um seine Arbeitsabläufe zu realisieren. Flußdiagramme beinhalten grundsätzlich mehrere Module mit Daten, Programmen und Ergebnissen. Mindestens eines dieser Module enthält ein vom Anwender erstelltes MI-STRAL-Programm, mit dem in der Regel aus einem oder mehreren Eingangsmodulen die Daten für die Ausgangsmodule erzeugt werden.

Die Flußdiagramme im UMDBS sind optimal in die übliche Abfolge der Arbeitsschritte angepaßt und erleichtern damit auch dem ungeübten Benutzer die Arbeit. Die notwendigen Arbeitsschritte sind jeweils über Module verdeutlicht, die über Pfeile in logischer Abfolge miteinander verbunden sind.



Module, die unbedingt benötigt werden, sind zu Beginn rot hinterlegt. Wird der Inhalt eines Moduls eingelesen oder editiert, wird (beim Einlesen automatisch, nach dem Editieren über den Befehl "Initialisieren") das enthaltene Programm bzw. die Datenstruktur analysiert. Bei korrektem Inhalt wird das entsprechende Modul anschließend grün hinterlegt. Ansonsten wird ein Fehler angezeigt, der in der Regel über einen sich automatisch öffnenden Editor direkt an der fehlerhaften Stelle im Programmcode behoben werden kann.

Module, die optional sind, werden zu Beginn grau dargestellt. Wird bei der Initialisierung eines Moduls festgestellt, daß Daten eines optionalen anderen Moduls benötigt werden, wird dieses anschließend automatisch rot dargestellt, um die Notwendigkeit für dessen Bearbeitung zu zeigen.

Die Pfeile zwischen den Modulen werden ebenfalls in den Farben Rot, Grün und Grau dargestellt. Damit wird verdeutlicht, ob diese Module konsistent zueinander sind. Werden z.B. in sich korrekte Parameter- und Simulationsmodelle initialisiert, die jedoch eine nicht zueinander passende Typendefinition enthalten, so werden beide Module grün dargestellt, der Verbindungspfeil zwischen ihnen jedoch rot. Die Konsistenzprüfung wird automatisch durchgeführt, sobald eines der beteiligten Module seinen Zustand ändert (z.B. beim Initialisieren).

Die eigentliche Aufgabe innerhalb eines Flußdiagramms kann dann gestartet werden, wenn keine roten Module oder Pfeile mehr vorhanden sind. In diesem Fall werden auch die Pfeile zu den Ergebnismodulen grün dargestellt.

Nahezu alle Befehle innerhalb von Flußdiagrammen lassen sich am schnellsten über Kontext-Menüs aufrufen, die zu jedem Modul individuell verfügbar sind. Zusätzlich läßt sich der jeweils wichtigste Befehl direkt durch einen Doppelklick auf das Modul aktivieren. Dieser Befehl hängt vom aktuellen Zustand des Moduls ab und ändert sich daher. So wird z.B. bei einem Doppelklick auf ein leeres Modul zunächst der Befehl "Laden..." aufgerufen. Ist das Modul dann geladen und initialisiert, wird - je nach Art des Moduls - ein Monitor oder ein Informationstext geöffnet. Ist das gesamte Flußdiagramm initialisiert, wird die Simulation durch Doppelklick auf das Simulationsmodell gestartet.

Da viele der Module in mehreren Flußdiagrammen enthalten sind, werden in diesem Kapitel zunächst die einzelnen Module mit ihrer spezifischen Funktion vorgestellt. Anschließend erfolgt die Erläuterung der Flußdiagramme selbst.

## 3.2 Module

### 3.2.1 Mikrodatenbasis

Die Mikrodatenbasis enthält die Werte der Attribute von Mikrodaten. Diese können innerhalb einer Datei unterschiedlichen Klassen angehören, die in der Regel zusammenwirken (z.B. Personen und Haushalte). Die Mikrodatenbasis enthält die Typ- und Klassendefinition, wie sie z.B. auch im Simulationsprogramm (in MISTRAL geschrieben) enthalten ist.

Die Mikrodatenbasis wird normalerweise über das Flußdiagramm Generierung/Import von externen Dateien in das System eingelesen und steht dann dort zur Verfügung. Durch die Simulation wird die Mikrodatenbasis verändert und kann anschließend abgespeichert werden.

Sofern ein manueller Eingriff in die Datenbasis notwendig wird oder die Datei von einem externen System erzeugt werden soll, kann auf die Beschreibung der Syntax innerhalb des MISTRAL-Handbuchs zurückgegriffen werden.

Die Mikrodatenbasis wird in Dateien namens \*.odb (object database) abgelegt.

### 3.2.2 Simulationsmodell

Das Simulationsmodell wird vom Benutzer in der Sprache MISTRAL geschrieben und ist mit einem PASCAL-Programm vergleichbar. Der wesentliche Vorteil von MISTRAL besteht jedoch darin, daß - neben einer Reihe von Spezialbefehlen für die Mikrosimulation - vor allem die Schnittstellen zu den benötigten anderen Modulen (Mikrodatenbasis, Parameter, Report usw.) vom System automatisch realisiert werden.

Das Simulationsmodell wird mit dem Schlüsselwort "SIMULATION" eingeleitet und in Dateien namens \*.mml (micro modelling language) abgespeichert.

### 3.2.3 Parameterdatenbasis

Die Parameterdatenbasis enthält eine Vielzahl von Parametern für Generierung und Simulation, die in Tabellen mit Subtabellen strukturiert sind (siehe dazu das MISTRAL-Handbuch). Obwohl es sich weitgehend um Daten anstelle von Algorithmen handelt, wird die Parameterdatenbasis im wesentlichen manuell vom Benutzer wie ein MISTRAL-Programm eingegeben und editiert.

Die Parameterdatenbasis wird in Dateien namens \*.pdb (parameter database) abgelegt.

### 3.2.4 Zeitreihendatenbasis

Die Zeitreihendatenbasis kann einerseits dazu verwendet werden, im Rahmen einer Mikrosimulation Makrowerte in die Simulation einzubinden (z.B. erwartete Tarifsteigerungen o.ä.), andererseits erzeugt eine Mikrosimulation in der Regel auch aggregierte Daten (z.B. Anzahl der Haushalte, Zahl der Geburten in einer Periode oder Summe aller Haushaltseinkommen), die bei einer Simulation über mehrere Perioden hinweg als Zeitreihen vorliegen. Die automatisch erzeugten Datenbasen besitzen dieselbe Syntax wie die manuell erzeugten und können auch in eigenen Datenbasen durch Kopieren auf Textebene weiter verwendet werden.

Die Parameterdatenbasis wird in Dateien namens \*.ts (time series) abgelegt.

### 3.2.5 Report

Der Report ist eine vom System beim Ablauf eines MISTRAL-Programms (z.B. bei der Simulation) automatisch erzeugte Textdatei, die ausschließlich der Information des Benutzers dient und vom UMDBS selbst nicht wieder eingelesen wird.

Der Report enthält immer einen Kopf mit Informationen zur verwendeten MISTRAL-Version und den Aktionen (z.B. "Simulation der Periode 1999"). Ansonsten wird der Inhalt des Reports weitgehend vom Benutzer durch Anweisungen innerhalb des MISTRAL-Programms bestimmt, mit denen einzelne Zeilen oder komplette Tabellen ausgegeben werden können.

Reports werden vom System automatisch mit der Erweiterung \*.rpt versehen, die jedoch geändert werden kann (z.B. in \*.txt für einen schnellen Zugriff mittels Texteditor).

### 3.2.6 Verteilungsdatenbasis

Die Verteilungsdatenbasis enthält Informationen, die mit dem MQL-Befehl DISTRIBUTION innerhalb eines Simulations- oder Analyseprogramms in MISTRAL erzeugt und an eine INTERFACE-Variable des Typs DISTRIBUTION übergeben wurden. Die Verteilungsdatenbasis ermöglicht damit eine einfache nachträgliche Analyse von Mikrodaten anhand vordefinierter Abfragen auch über mehrere Simulationsperioden.

Verteilungsdatenbasen werden ausschließlich vom System erzeugt und sind nicht für die manuelle Bearbeitung vorgesehen. Sie werden in Dateien namens \*.dis (distribution) abgelegt.

### 3.2.7 Analysemodell

Das Analysemodell enthält ein vom Benutzer geschriebenes MISTRAL-Programm, mit dem eine Mikrodatenbasis analysiert werden kann. Das Ergebnis dieser Analyse kann (parallel) als Text in einen Report, als Wert in eine Zeitreihendatenbasis und als Verteilung in eine Verteilungsdatenbasis ausgegeben werden.

Grundsätzlich könnten diese Anweisungen auch Teil eines Simulationsmodells sein. Die Auslagerung in ein getrenntes Analysemodell bietet jedoch eine Reihe von Vorteilen:

- Wiederkehrende Analysen können zusammen mit mehreren Simulationsmodellen verwendet werden, ohne daß diese bei Änderungen der Analysen jeweils angepaßt werden müßten.
- Die Analyse kann bei einer Simulation über mehrere Perioden zum einen die Ausgangsdatenbasis analysieren und im Anschluß an jede abgeschlossene Simulationsperiode das entstandene Ergebnis erneut analysieren.
- Die Analyse kann auch ohne Simulation direkt für die Untersuchung einer Mikrodatenbasis (z.B. nach Import) verwendet werden.
- Die klare Trennung von Programmcode mit unterschiedlichen Aufgaben entspricht den Regeln modernen Software-Engineerings und verbessert die Möglichkeiten einer Arbeit im Team sowie der Verwendung vorhandener Arbeitsergebnisse.

Das Analysemodell wird mit dem Schlüsselwort "ANALYSIS" eingeleitet und in Dateien namens \*.mml (micro modelling language) abgespeichert.

### 3.2.8 Testmodell

Ähnlich dem Analysemodell dient auch das Testmodell dazu, eine Analyseaufgabe sauber von z.B. Import oder Simulation zu trennen und universell verfügbar zu machen. Im Unterschied zur Analyse wird beim Test aber nur die Einhaltung von Regeln innerhalb einer Mikrodatenbasis mittels eines MISTRAL-Programms überprüft. Wird dabei ein Fehler festgestellt, ist als Ergebnis mittels WRITELN-Anweisung eine geeignete Fehlermeldung für den Report zu erzeugen.

Das Ergebnis des Tests besteht also ausschließlich in einem Text, der bei einer korrekten Mikrodatenbasis leer ist. Ein nicht leerer Text wird vom System als aufgetretener Fehler in der Mikrodatenbasis interpretiert und durch einen roten Rand im Testmodell signalisiert. Bei einer Simulation über mehrere Perioden kann optional festgelegt werden, daß die Simulation nach dem Entdecken eines Fehlers nach dem Abschluß einer Periode abbricht.

Das Testmodell wird mit dem Schlüsselwort "TEST" eingeleitet und in Dateien namens \*.mml (micro modelling language) abgespeichert.

### 3.2.9 Generierungsmodell

Das Generierungsmodell ähnelt sehr stark dem Simulationsmodell. Während dort jedoch die Fortschreibung einer vorhandenen Mikrodatenbasis um jeweils genau eine Periode (auch mehrfach hintereinander) vorgenommen wird, erzeugt das Generierungsmodell aufgrund unterschiedlicher Vorgaben eine neue Datenbasis. Dabei sind folgende Eingangsdaten möglich:

- Aus einer oder mehreren Textdateien werden Daten von externen Systemen importiert.

- Einzelne oder alle Attribute der Mikroobjekte sowie gegebenenfalls sogar die Mikroobjekte selbst werden - optional in Abhängigkeit von den Daten einer Parameterdatenbasis - aufgrund von Zufallswerten erzeugt.

Damit lassen sich unter anderem folgende Ziele realisieren:

- Mikrodatenbasen können aufgrund von Textdateien anderer Systeme eingelesen werden.
- Die Attribute der Objekte einer Mikrodatenbasis können erzeugt (z.B. MISSING\_VALUES), korrigiert (festgestellte Inkonsistenzen) oder transformiert (z.B. Transformation in eine andere Periode, Umrechnung von DM in Euro) werden.
- Mehrere Mikrodatenbasen bzw. Importdateien können zusammengefügt ("Mergen") werden (z.B. Personen und Haushaltsdaten oder Ost- und Westdeutschland).
- Eine Mikrodatenbasis kann in Ihrer Struktur ergänzt oder verändert werden (z.B. neue Attribute, andere Kodierung vorhandener Attribute)

Das Analysemodell wird mit dem Schlüsselwort "GENERATION" eingeleitet und in Dateien namens \*.mml (micro modelling language) abgespeichert.

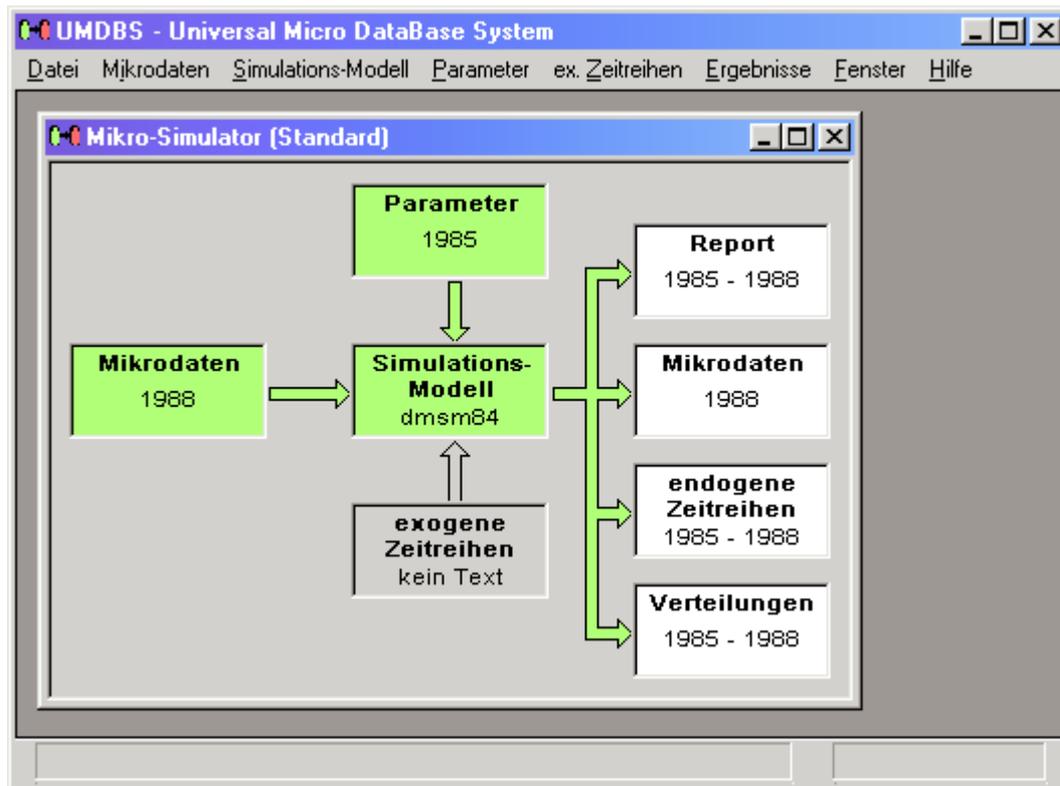
### 3.3 Simulation

Allen Simulations-Flußdiagrammen ist gemeinsam, daß eine Eingangs-Mikrodatenbasis mittels eines Simulationsmodells sowie optional einer Parameter- sowie Zeitreihendatenbasis in Schritten von jeweils einer Periode (in der Regel ein Jahr) fortgeschrieben wird. Als Ergebnis steht einerseits die veränderte Mikrodatenbasis für eine Analyse oder die Simulation der nächsten Periode zur Verfügung, andererseits werden weitere Ergebnisse in Form von Reports, Zeitreihen und Verteilungsdatenbasen erzeugt.

Neben dem Standard-Flußdiagramm gibt es eine erweiterte Form mit zusätzlichem Analyse- und Testmodell sowie eine Variante, die automatisch eine große Anzahl von Simulationsläufen mit veränderten Zufallszahlen startet und für eine statistische Auswertung zusammenfaßt.

#### 3.3.1 Standard-Simulation

Das Flußdiagramm der Standard-Simulation ist in nachfolgendem Bildschirmausdruck zu sehen:



Auf der linken Seite befindet sich die Mikrodatenbasis, die Mikroobjekte einer bestimmten Periode repräsentiert. In der Mitte ist das Simulationsmodell mit dem in MISTRAL formulierten Fortschreibungsalgorithmus enthalten. Darüber befindet sich die optionale, aber fast immer verwendete Parameterdatenbasis mit den Parametern und Wahrscheinlichkeiten für die Simulation. Die seltener verwendete Zeitreihendatenbasis darunter dient dazu, exogene aggregierte Größen (z.B. Tarifabschlüsse usw.) für die Simulation zur Verfügung zu stellen.

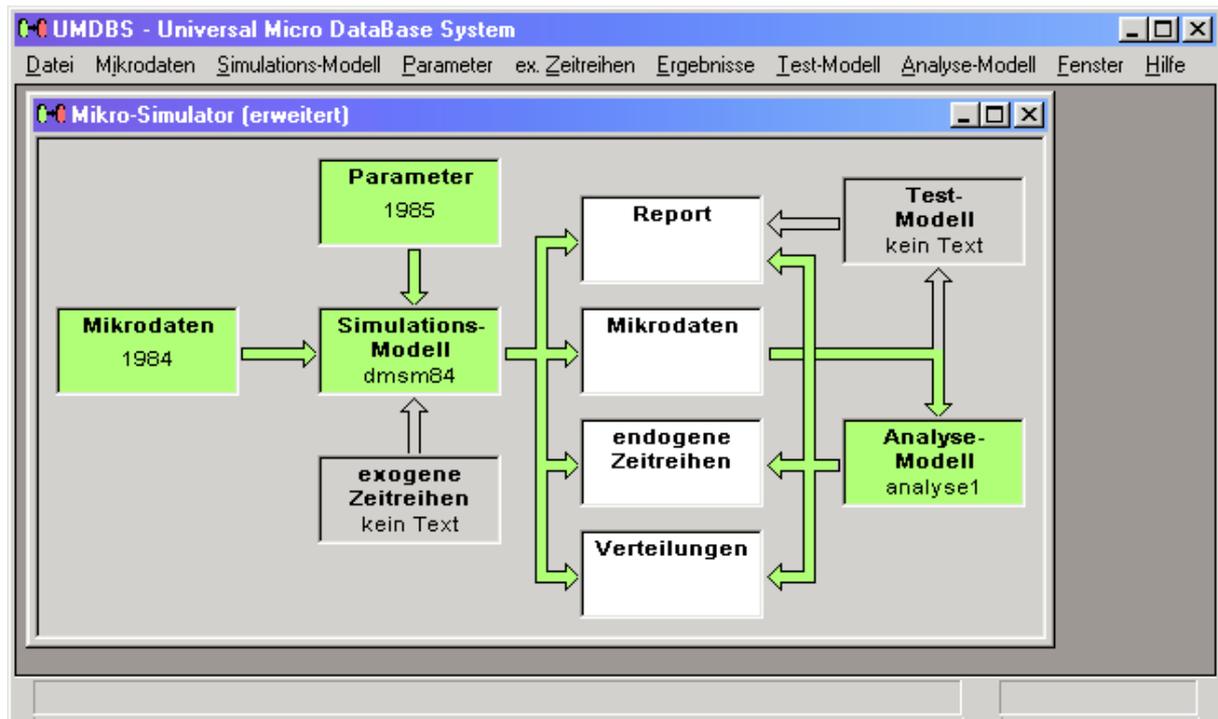
Auf der rechten Seite sind die Module zu sehen, die die Ergebnisse der Simulation aufnehmen:

Die Mikrodatenbasis enthält die Mikrodaten, die sich durch Fortschreibung der Eingangsdatenbasis ergeben haben. Da keine Kopie der Ursprungsdaten angelegt wird und das Ergebnis wiederum als Eingangsdaten für die Simulation der nächsten Periode verwendet wird, ist die Mikrodatenbasis rechts und links des Simulationsmodells identisch. Lediglich aus der Logik der Anwendung heraus stehen in den jeweiligen Menüs unterschiedliche Befehle zur Verfügung.

Der Report enthält Systeminformationen sowie die Textausgaben des Simulationsmodells. Die Zeitreihendatenbasis "endogene Zeitreihen" enthält die aggregierten Simulationsergebnisse (ein Wert pro Größe und Periode); die Verteilungsdatenbasis umfaßt die disaggregierten Ergebnisse, die vom MQL-Befehl DISTRIBUTION stammen.

### 3.3.2 Erweiterte Simulation

Das Flußdiagramm der erweiterten Simulation präsentiert sich folgendermaßen:



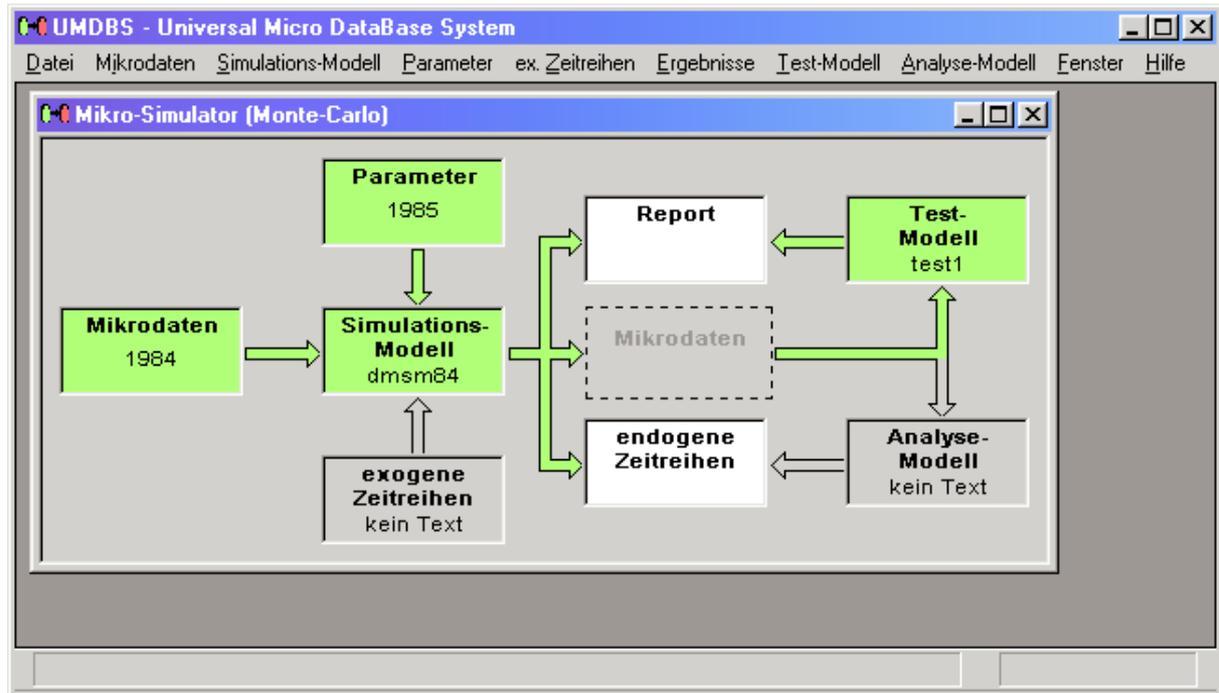
In Ergänzung zu den bisherigen Modulen wurden jeweils ein Test- und ein Analysemodul hinzugefügt. Beide dienen dazu, die durch die Simulation erzeugte Mikrodatenbasis zu untersuchen. Optional kann dies auch einmalig vor dem Start der Simulation für die von einer Datei geladene Mikrodatenbasis durchgeführt werden.

### 3.3.3 Monte-Carlo-Simulation

Die Ergebnisse jeder Mikrosimulation als stochastischer Simulation unterliegen einer unvermeidlichen Zufallsstreuung, die auch als Monte-Carlo-Streuung bezeichnet wird. Trotz der vermeintlich großen Datenmengen von meist mehreren tausend Haushalten liegt die Streuung für die meisten Ergebnisgrößen im - teils sogar zweistelligen - Prozentbereich. Es ist deshalb unabdingbar, mehrere Simulationen mit jeweils anderen Zufallszahlen durchzuführen und die Ergebnisse zu mitteln. Damit wird einerseits die Streuung verringert, zum anderen wird so eine Aussage über den mit der Streuung verbundenen Simulationsfehler möglich.

Das UMDBS stellt über das Flußdiagramm "Monte-Carlo-Simulation" die Möglichkeit zur Verfügung, eine vorher eingestellte Anzahl (z.B. 30) von Simulationsläufen (optional jeweils über mehrere Perioden) automatisch mit jeweils anderen Zufallszahlen durchzuführen. Dies geschieht in der Weise, daß die Ausgangsdatenbasis zunächst als Kopie intern gesichert wird. Anschließend wird eine normale Simulation über die gewünschte Anzahl von Perioden durchgeführt. Nach dem Ende eines Simulationslaufs wird die anfangs gesicherte Mikrodatenbasis kopiert, diese Kopie mit neuen Zufallszahlen initialisiert und als Ausgangspunkt eines weiteren Simulationslaufs über dieselben Perioden wie zuvor verwendet. Nach Abschluß aller Durchläufe kann das Ergebnis mit dem Monte-Carlo-Monitor untersucht werden.

Das Flußdiagramm sieht folgendermaßen aus:



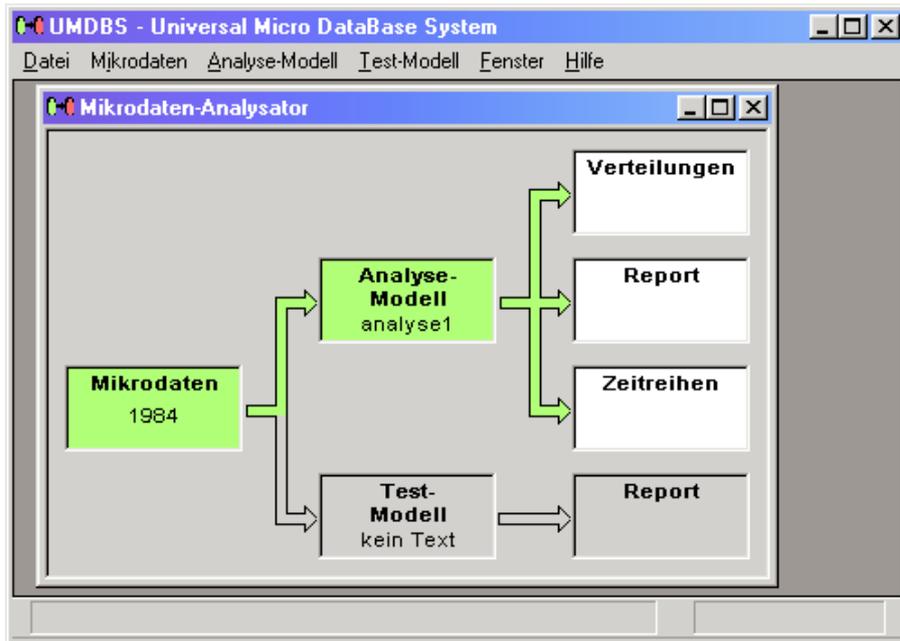
Es entspricht weitgehend dem der erweiterten Simulation. Prinzipbedingt fehlen allerdings einige Ergebnisse:

Die Mikrodatenbasis steht hier nicht zur Verfügung, da sie jeweils wieder für den nachfolgenden Durchlauf gelöscht wurde. Auch die Verteilungsdatenbasis ist hier nicht sinnvoll, da für jede Größe und Periode z.B. 30 verschiedene Ergebnisse berechnet wurden.

Neben dem Report, der hier jedoch besser keine umfangreichen Ausgaben des Simulationsmodells enthalten sollte, stellt die Zeitreihendatenbasis das wesentliche Ergebnis dar. Sie enthält für jede (aggregierte) Größe und jeden Durchlauf eine getrennte Zeitreihe. Diese Daten können dann mit einem Monte-Carlo-Monitor komfortabel ausgewertet werden.

### 3.4 Analyse

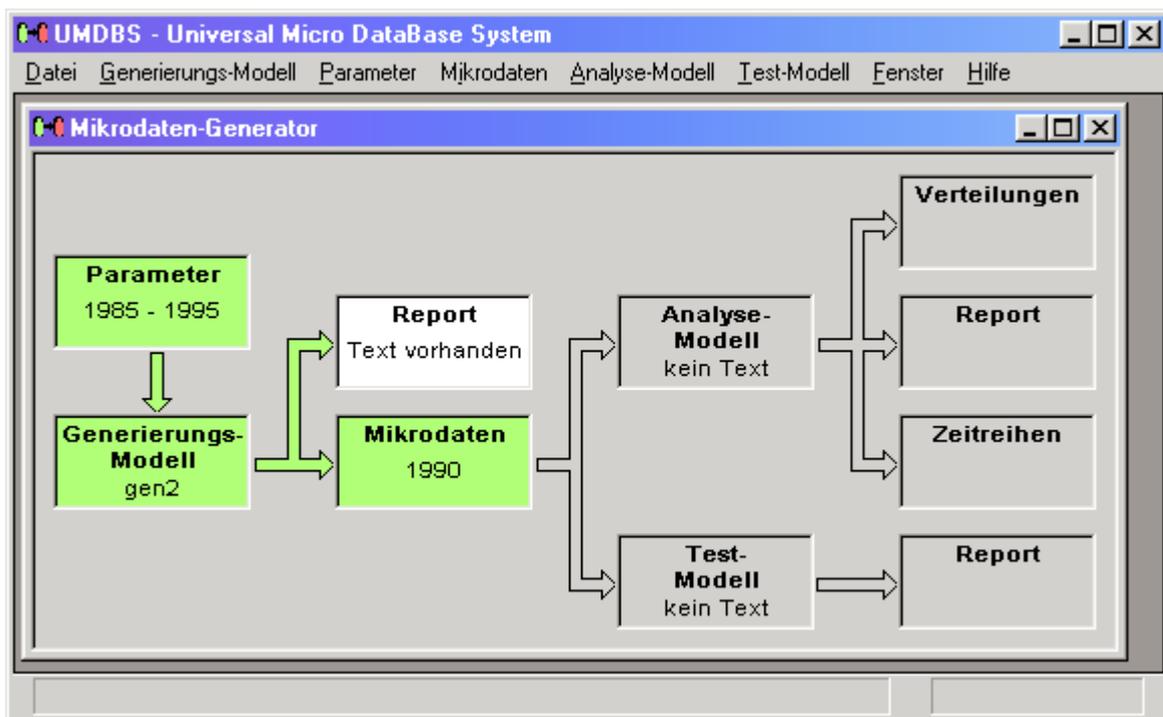
Das Flußdiagramm zur Analyse dient vor allem dazu, Mikrodatenbasen statisch (d.h. ohne Simulation) zu untersuchen. Dazu stehen ein Test- und ein Analysemodul mit den jeweils passenden Ergebnismodulen zur Verfügung:



Nach dem Einlesen der Mikrodatenbasis können ein oder auch beide nachfolgenden Module initialisiert und nacheinander manuell gestartet werden.

### 3.5 Generierung / Import

Mit dem Flußdiagramm "Generierung/Import" können Mikrodatenbasen aus verschiedenen Quellen erzeugt werden (vgl. Abschnitt 3.2.9). Aktuell sind eine Parameterdatenbasis sowie bis zu 15 Textdateien (ohne sichtbare Module) möglich. Künftig sollen zusätzlich bis zu fünf Mikrodatenbasen als Eingangsdaten verwendet werden. Der rechte Teil des Flußdiagramms entspricht dem oben dargestellten Analyse-Flußdiagramm.



Die Namen der Input-Textdateien können einerseits innerhalb des Generierungsmodells direkt angegeben werden. Flexibler ist jedoch die Möglichkeit, diese Angaben über ein Dialogfeld vorzunehmen. Wurde kein Dateiname spezifiziert, wird das entsprechende Dialogfeld automatisch beim Initialisieren des Moduls bzw. beim Start der Generierung geöffnet.

## 4 Editoren

Mit Editoren können Programme und Datenbestände eingegeben und verändert werden.

### 4.1 Textbasierende Editoren

Die textbasierende Editoren innerhalb des UMDBS dienen dazu, MISTRAL-Programmcode als ASCII-Text einzugeben und zu editieren. Dazu stellen sie praktisch alle heute dafür üblichen Funktionen zur Verfügung. Besonders benutzerfreundlich ist die direkte Integration in den Compilationsprozeß. Wird vom Compiler ein Syntax- oder Typ-Fehler festgestellt, öffnet sich automatisch der Texteditor mit dem Programmcode, plaziert den Cursor an der fehlerhaften Stelle und zeigt im Statusbereich die Fehlerposition und -art an:

The screenshot shows a window titled "UMDBS - Universal Micro DataBase System" with a menu bar (Datei, Bearbeiten, Modell, Fenster, Hilfe). Inside, a sub-window titled "MML-Modell-Editor (Simulations-Modell) für Mikro-Simulator" displays the following code:

```
(Alter: INTEGER;
Nationalitaet: Nationalitaeten;
Familienstand: Familienstaende;
r: PROB): Berufe;
Arbeitslosigkeit
(Alter: INTEGER;
Nationalitaet: Nationalitaeten;
Geschlecht: Geschlechter;
Beruf: Berufe): PROB;

VAR
P Person;
Partnermarkt: SET OF Person;

#####
# Hilfs-Routinen #
#####

### Zentraler Partner-Markt ###
```

The status bar at the bottom indicates a syntax error: "nicht initialisiert Fehler in Zeile 142, Spalte 5" and "Cursor-Position: Zeile 142 Spalte 5". The error message "":" erwartet." is highlighted in red.

Nach dem Korrigieren des Fehlers kann das Programm sofort gespeichert und neu übersetzt werden.

Es gibt z.Z. drei Varianten textbasierender Editoren:

- für MML-Modelle (Simulation, Analyse, Test und Generierung)
- für Parameterdatenbasen
- für Zeitreihendatenbasen

Alle diese Editoren besitzen ein einheitliches Menü "Bearbeiten", mit dem alle üblichen Funktionen zur Textbearbeitung aufgerufen werden können. Aufgrund interner Beschränkung ist die Funktion "Rückgängig" nur rudimentär vorhanden und hilft vor allem, versehentlichen Datenverlust durch unbeabsichtigtes Überschreiben beim Kopieren zu vermeiden.

Um das bei Programmtexten sinnvolle Einrücken zu unterstützen, werden nach einem <Return> automatisch so viele Leerzeichen wie in der Zeile davor eingefügt.

Unter dem ebenfalls überall verfügbaren Menü "Modell" besteht jeweils die Möglichkeit, den gerade edierten Text unmittelbar zu kompilieren ("Initialisieren"). Es ist dabei zu beachten, daß damit nur das einzelne Modell initialisiert wird, während dieselbe Funktion aus dem Flußdiagramm heraus aufgerufen bei erfolgreicher Initialisierung des einzelnen Moduls anschließend auch die Verbindung zu den übrigen Modulen initialisiert. Nach dem Schließen des Editors ist deshalb gegebenenfalls eine nochmalige Initialisierung notwendig.

Unter dem Menü "Modell" stehen ferner Analyse-Funktionen zur Verfügung, die von der jeweiligen Modellart abhängen. Einheitlich ist der Menüpunkt "Info-Text", der einen kurz gefaßten Informationstext in einem zusätzlichen Fenster anzeigt. Beim Parameter- und Zeitreihen-Editor können zudem Monitore aufgerufen werden. Alle Analyse-Funktionen sind nur nach erfolgreicher Initialisierung des Moduls verfügbar.

In allen Editoren stehen im Menü "Datei" Funktionen zum komfortablen **Arbeiten mit Include-Dateien** zur Verfügung. Um direkt aus dem Quelltext eine einzelne Include-Datei zu öffnen, wird der Text-Cursor in der Zeile mit #INCLUDE plaziert und die Datei mit "Aktuelle Include-Datei öffnen" in einem zusätzlichen Editor geöffnet. Ist das Modell erfolgreich initialisiert worden, kann mit "Include-Datei öffnen..." eine beliebige untergeordnete Include-Datei in einem Dialogfenster aus einer hierarchisch angeordneten Liste ausgewählt werden. Mit dem Befehl "Alle Include-Dateien einbinden..." wird das über mehrere, auch hierarchisch angeordnete Include-Dateien aufgeteilte Programm zu einem einzigen Text im Hauptprogramm zusammengefaßt. Die bisherigen Anweisungen #INCLUDE werden dabei entfernt.

Sowohl der Parameter- als auch der Zeitreihen-Editor besitzen ein weiteres Menü "Generieren".

Im **Parameter-Editor** stehen umfangreichere Generierungsfunktionen zur Verfügung, die beim erstmaligen Erstellen des Codes hilfreich sind:

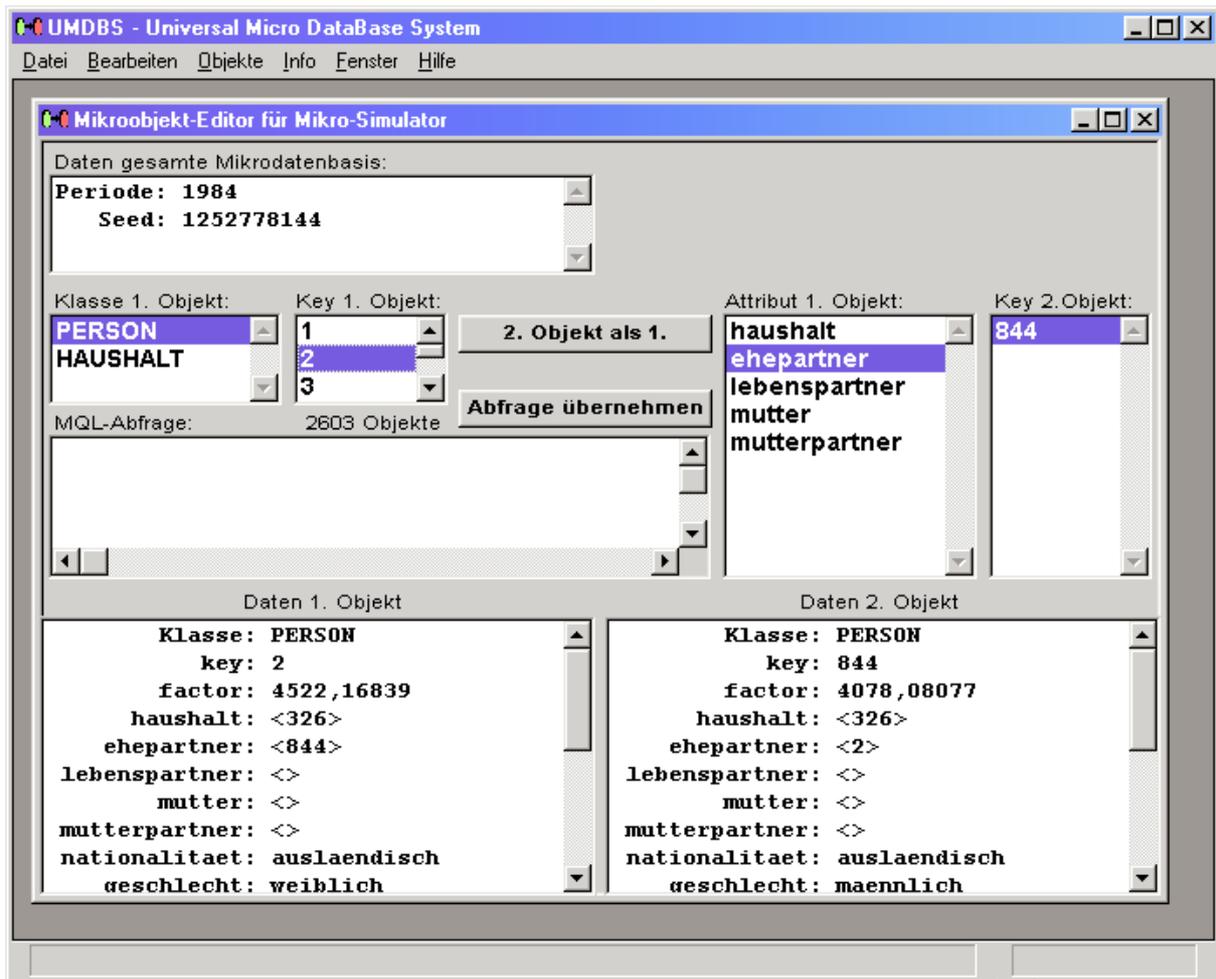
Mit "Tabellenrumpf generieren..." kann die Grundstruktur einer Tabelle inkl. ihrer Subtabellen automatisch erzeugt werden. Die notwendigen Vorgaben werden dazu über ein Dialogfenster abgefragt.

Sollen Daten aus anderen Programmen bzw. Quellen in Subtabellen übernommen werden, kann der Befehl "Inhalte einfügen..." wertvolle Hilfe leisten. Dazu müssen die Ursprungsdaten zunächst in die Zwischenablage von Windows kopiert werden. Dies kann in einem beliebigen anderen Programm (z.B. Excel) oder auch in einem Texteditor innerhalb des UMDBS mit "Kopieren" erfolgen. Nach Aufruf des Befehls können in einer Dialogbox Vorgaben für den Datenimport vorgenommen werden. Z.B. können störende 1000er Trennzeichen entfernt und der nationale Trenner "," für die Nachkommastellen in den benötigten Punkt umgewandelt werden. Besonders wichtig ist die Angabe der Dimensionen, mit denen die für die Parameter benötigten Klammerebenen automatisch erzeugt werden. Bei Einlesen mit "Inhalte einfügen..." bleiben alle Teile des Ursprungstextes unberücksichtigt, die keine Zahlen sind. Damit können bequem komplette Tabellen inkl. Überschriften und beliebigen Trennzeichen importiert werden.

Beim **Zeitreihen-Editor** kann dazu der gesamte Quellcode des initialisierten Moduls vollständig neu erzeugt werden (ähnlich einem "Pretty Printing"), wobei vom Benutzer über den Menüpunkt "Optionen" dazu Vorgaben gemacht werden können. Auch hier gibt es mit "Inhalte einfügen..." eine sehr komfortable Funktion zum Import aus anderen Programmen.

## 4.2 Mikroobjekt-Editor

Für das unmittelbare Bearbeiten einzelner Mikroobjekte ist der Mikroobjekt-Editor gedacht, der weitgehend dem Mikroobjekt-Monitor entspricht:



Im Teilfenster links oben sind Eigenschaften der gesamten Mikrodatenbasis aufgelistet. Zur Zeit sind dies die Angabe der Periode, die die Mikrodatenbasis repräsentiert, sowie ein Startwert ("Seed") für den internen Zufallsgenerator, aus dem heraus beim Einlesen die individuellen Zufallsgeneratoren alle Mikroobjekte initialisiert werden.

Darunter können einzelne Mikroobjekte über ihre Klasse und Schlüsselnummer (Key) ausgewählt und mit Ihren Attributen angezeigt werden. Attribute, die auf andere Objekte (auch anderer Klassen) zeigen, werden rechts aufgelistet. Die Attribute dieser Objekte können parallel im rechten unteren Bereich des Fensters dargestellt werden. Sollen gezielt Mikroobjekte mit bestimmten Eigenschaften editiert werden, kann diese Teilmenge über MQL-Befehle im Textfenster ("MQL-Abfrage:") bestimmt werden (nach Eingabe auf "Abfrage übernehmen" klicken).

Um einzelne Attribute zu ändern, wird per Doppelklick darauf ein Dialogfenster geöffnet, in dem die zulässigen Ausprägungen aufgelistet werden. Über das Menü "Objekte" können einzelne Objekte auch vollständig aus der Mikrodatenbasis entfernt werden. Alle Zeiger auf das entfernte Objekt werden dabei automatisch auf NIL gesetzt; außerdem wird das Objekt aus allen SETs entfernt. Es ist zu beachten, daß z.B. mit einem Haushalt nicht automatisch auch die darin befindlichen Personen gelöscht werden, da diese eigenständige Objekte darstellen, die weiterhin in der Liste der Klasse "Person" o.ä. auftauchen. Soll ein Haushalt vollständig inkl. aller Personen gelöscht werden, sollte im linken Teilfenster der Haushalt, im rechten dessen Personen dargestellt werden. Dann werden zunächst alle Personen (Objekt 2) gelöscht; anschließend der dann leere Haushalt.

## 5 Monitore

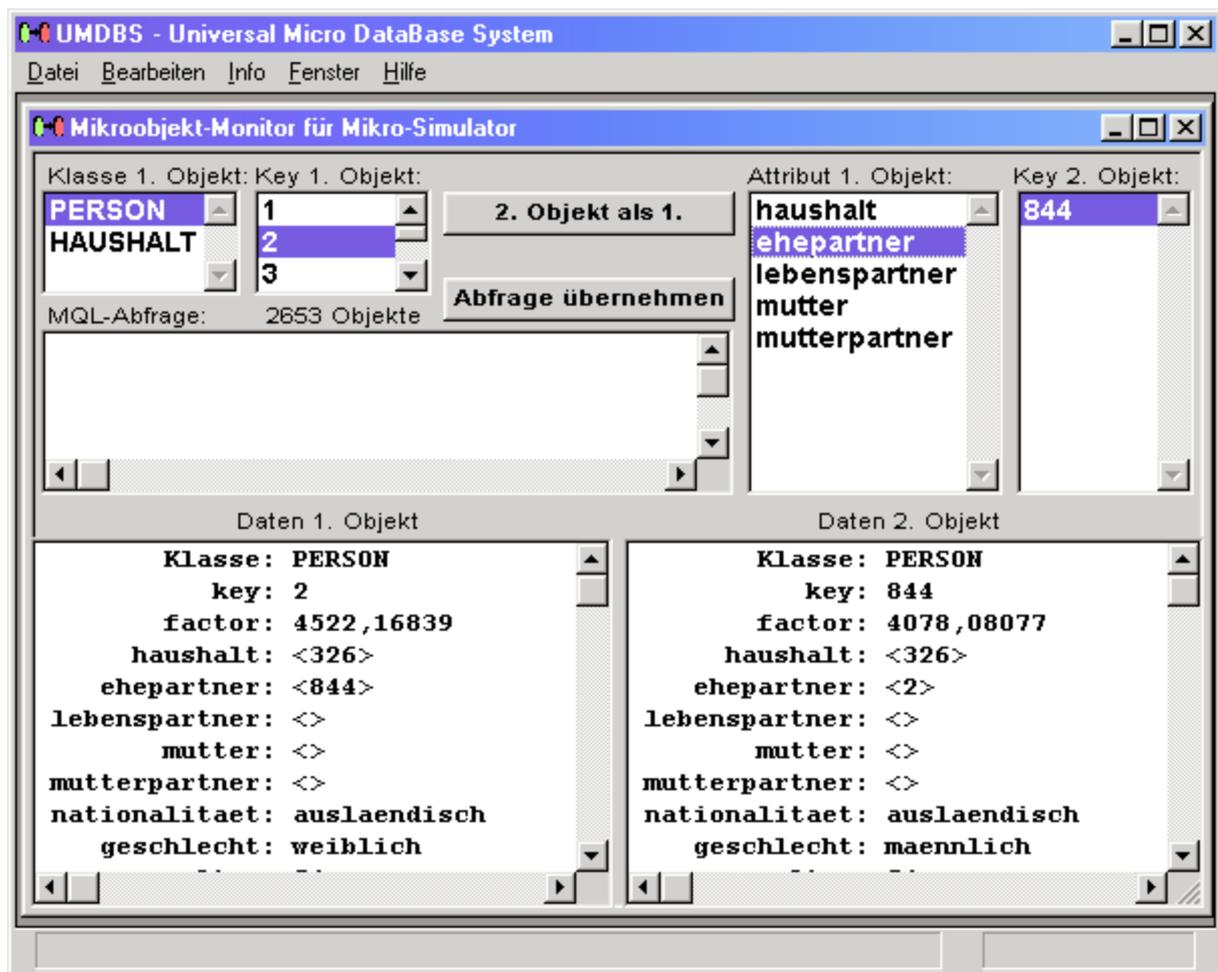
Monitore dienen dazu, Datenbestände interaktiv zu analysieren. Je nach Art des konkreten Monitors stehen dabei grafische und/oder tabellarische Anzeigen zur Auswahl. Die Anzeige kann in der Regel zusätzlich über Menübefehle und Dialogfenster beeinflusst werden.

Für die verschiedenen Arten von Daten stehen innerhalb des UMDBS unterschiedliche Monitore zur Verfügung:

- Mikrodatenbasen: Mikroobjekt-Monitor, Mikro-Monitor
- Verteilungsdatenbasen: Verteilungs-Monitor
- Zeitreihendatenbasen: Zeitreihen-Monitor, Makro-Monitor
- Monte-Carlo-Ergebnisse: Monte-Carlo-Monitor
- Parameterdatenbasen: Parameter-Monitor

### 5.1 Mikroobjekt-Monitor

Der Mikroobjekt-Monitor dient dazu, die Objekte einer Mikrodatenbasis einzeln mit ihren Attributen darzustellen. Die Benutzeroberfläche sieht so aus:



Links oben wird zunächst die Klasse ausgewählt, deren Objekte untersucht werden sollen. Im Listenfeld rechts davon die Schlüsselnummern dieser Objekte angezeigt. Dort kann ein Objekt angeklickt oder auch (über das Popup-Menü) anhand seines Schlüssels direkt gesucht werden. Anschließend werden links unten die Attribute des gewählten Objekts im Klartext angezeigt.

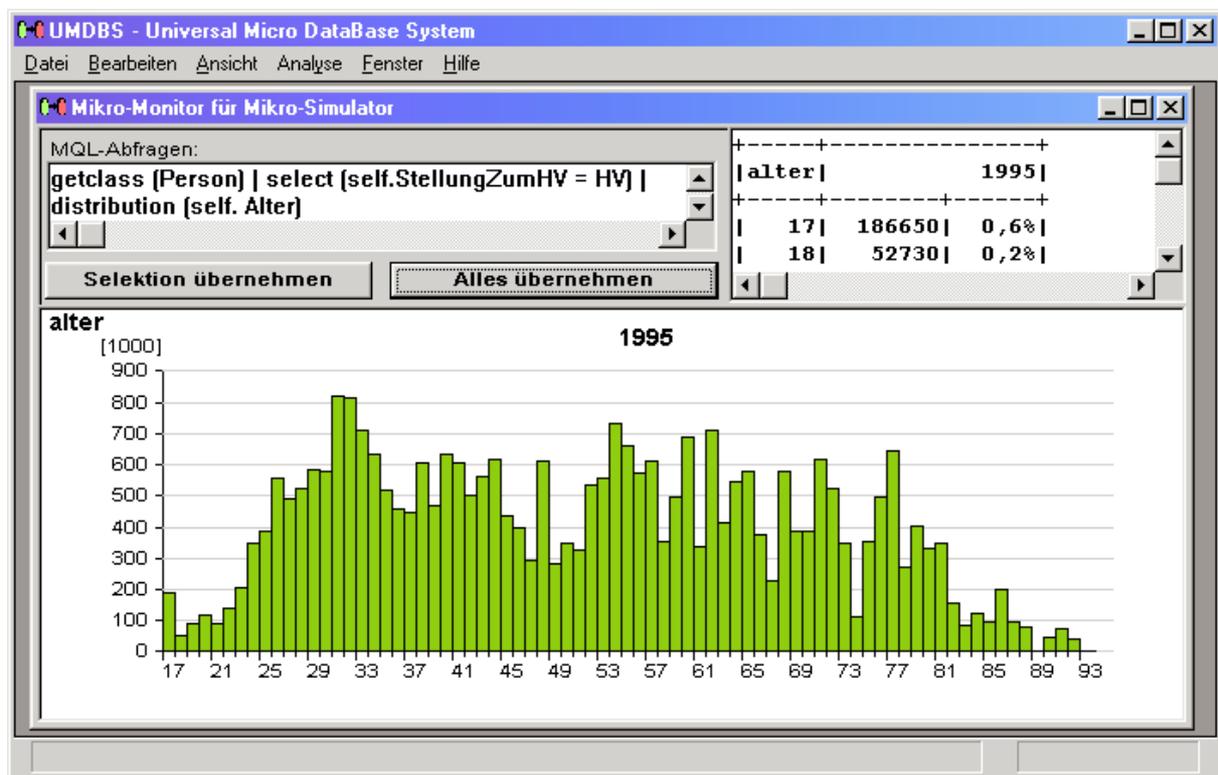
Im zweiten Listenfeld von rechts oben werden die Attribute des gewählten Objekts (bzw. seine Klasse) angezeigt, die Zeiger auf andere Objekte (auch anderer Klassen) darstellen. Wird eines dieser Attribute angeklickt, erscheinen die Schlüssel der entsprechenden Objekte rechts davon. Bei SETs kann dann eines davon für die Darstellung unten rechts ausgewählt werden; handelt es sich nur um einen Zeiger auf ein einzelnes Objekt, erscheint dieses automatisch.

Die parallele Darstellung zweier verbundener Objekte erlaubt eine rasche Kontrolle der Konsistenz von Daten (z.B. gleiche Ehedauer bei Ehepartnern). Weiterhin wird damit auch ein schnelles Navigieren durch die Mikrodatenbasis erleichtert, da mit der entsprechenden Befehlsschaltfläche ("2. Objekt als 1.") das rechte Objekt als Hauptobjekt übernommen werden kann. Damit kann z.B. von einem Mann auf dessen Lebenspartnerin und von einer Person auf deren Haushalt übergegangen werden.

Soll gezielt eine bestimmte Teilgruppe von Objekten untersucht werden, kann durch Eingabe einer MQL-Abfrage (ohne Abschlußbefehl) eine entsprechende Definition vorgenommen werden und mit der Schaltfläche "Abfrage übernehmen" aktiviert werden. Z.B. liefert "getclass (Person) | select (self.Alter = 0)" eine Liste aller Neugeborenen.

## 5.2 Mikro-Monitor

Der Mikro-Monitor dient dazu, interaktiv eine Mikrodatenbasis bezüglich ihrer Verteilungsinformationen zu analysieren:



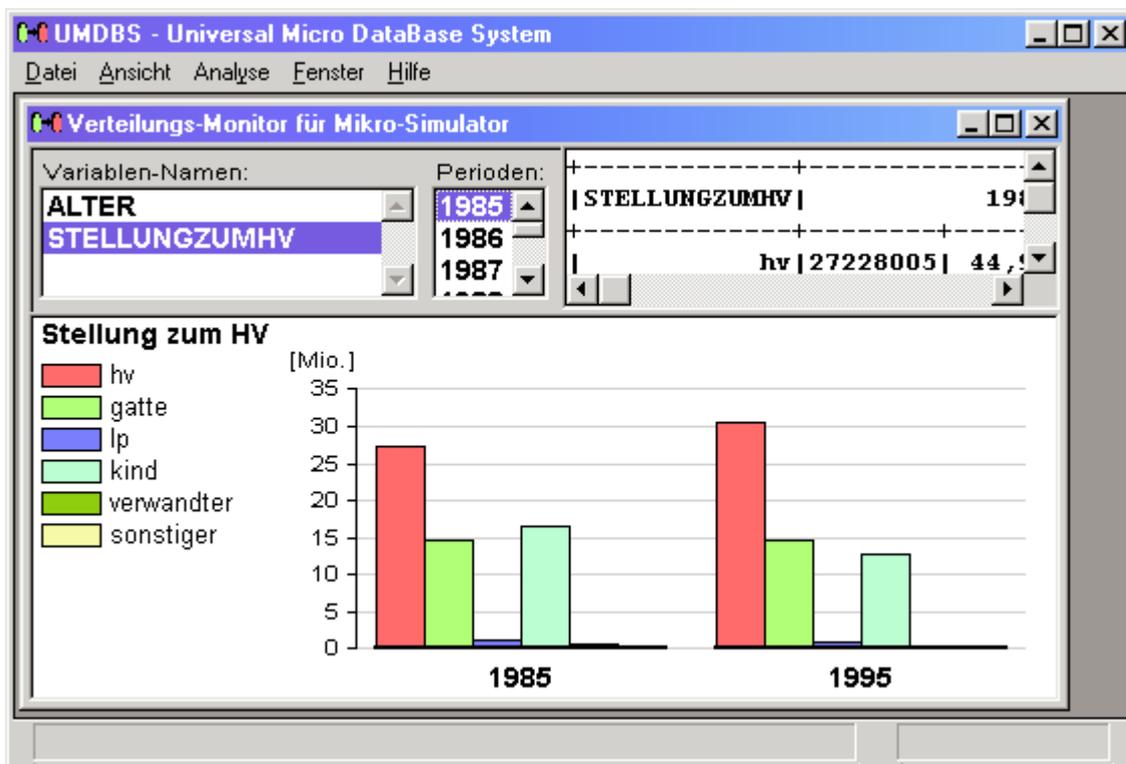
Links oben befindet sich ein Texteditor-Fenster, in dem der Benutzer MQL-Abfragen eingeben kann. Befindet sich dort nur jeweils eine MQL-Abfrage, so wird sie mit "Alles übernehmen" aktiviert. Ansonsten kann aus einem umfangreicheren Text auch ein Teil selektiert und mit "Selektion übernehmen" eingelesen werden.

Das Ergebnis einer solchen Abfrage (meist eine Verteilung) wird im unteren Bereich grafisch sowie rechts oben als Tabelle dargestellt. Die Darstellung richtet sich nach der Art der Daten. So werden metrische eindimensionale Daten automatisch als Histogramm, metrische zweidimensionale Daten als Streudiagramm angezeigt. Für qualitative Daten (nominal und ordinal) stehen Kreisdiagramm, Säulendiagramm und Stapelsäulendiagramm zur Auswahl.

Über zahlreiche Menü-Befehle und Optionen kann die Anzeige den eigenen Wünschen entsprechend beeinflusst werden. Zusätzlich können über das Menü "Analyse" einige wichtige statistische Kennzahlen angezeigt werden.

### 5.3 Verteilungs-Monitor

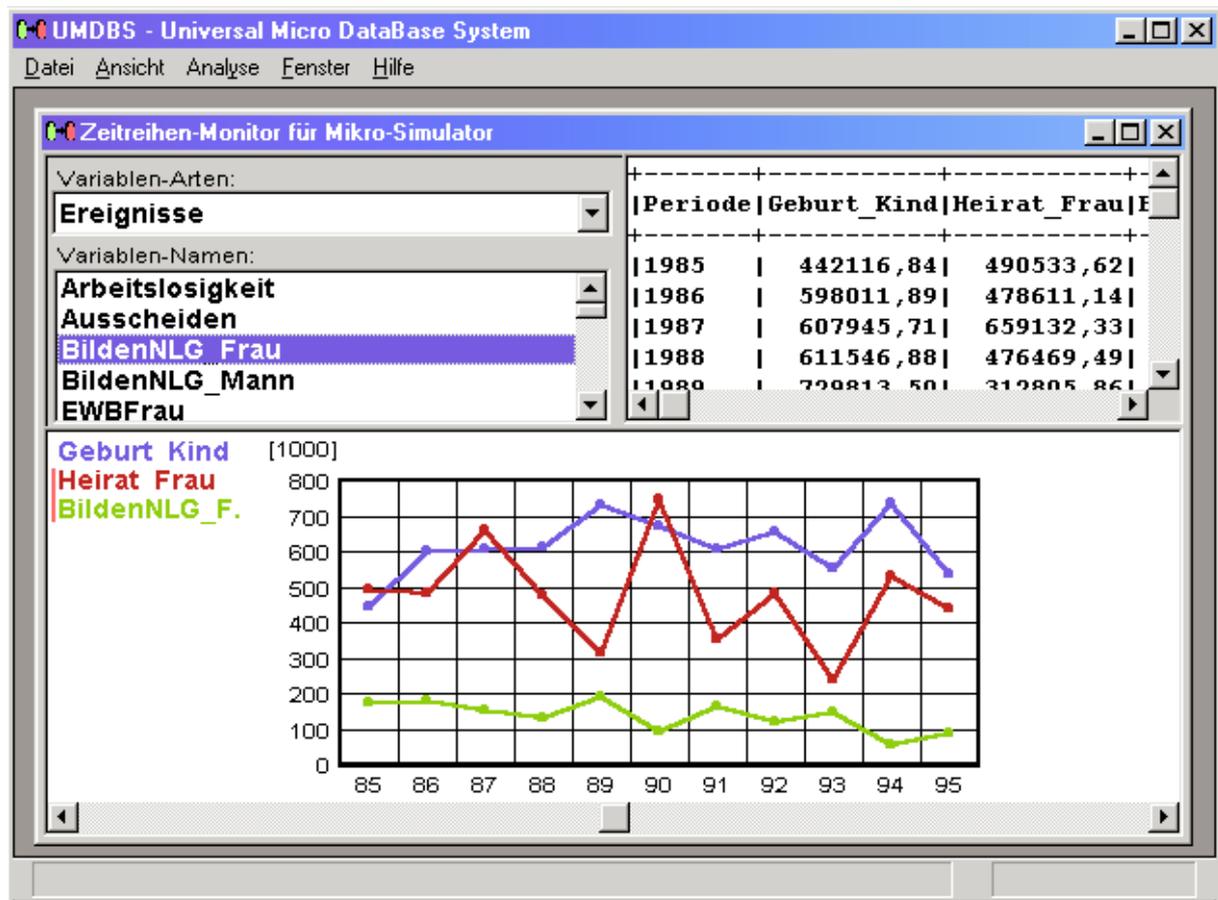
Der Verteilungs-Monitor ähnelt sehr stark dem Mikro-Monitor, dient jedoch nicht der interaktiven Analyse einer Mikrodatenbasis, sondern der Analyse einer Verteilungsdatenbasis, die im Rahmen einer Simulation entstanden ist. Deshalb sind keine Eingaben von MQL-Befehlen möglich. Statt dessen werden links oben die vom Simulationsprogramm erzeugten Variablen sowie die verfügbaren Perioden angezeigt und können durch Anklicken ausgewählt werden.



Der Vorteil des Verteilungs-Monitors gegenüber dem Mikro-Monitor besteht darin, daß die immer wieder neue Eingabe von Abfragen entfällt und zudem mehrere Perioden zur vergleichenden Analyse nebeneinander dargestellt werden können. Andererseits sind aber keinerlei nachträgliche Ergänzungen oder Änderungen der Abfragen (außer z.B. andere Klassenbildung) mehr möglich.

## 5.4 Zeitreihen-Monitor

Der Zeitreihen-Monitor erlaubt die grafische und tabellarische Darstellung des Inhalts von manuell erstellten oder über die Simulation erzeugten Zeitreihendatenbasen.

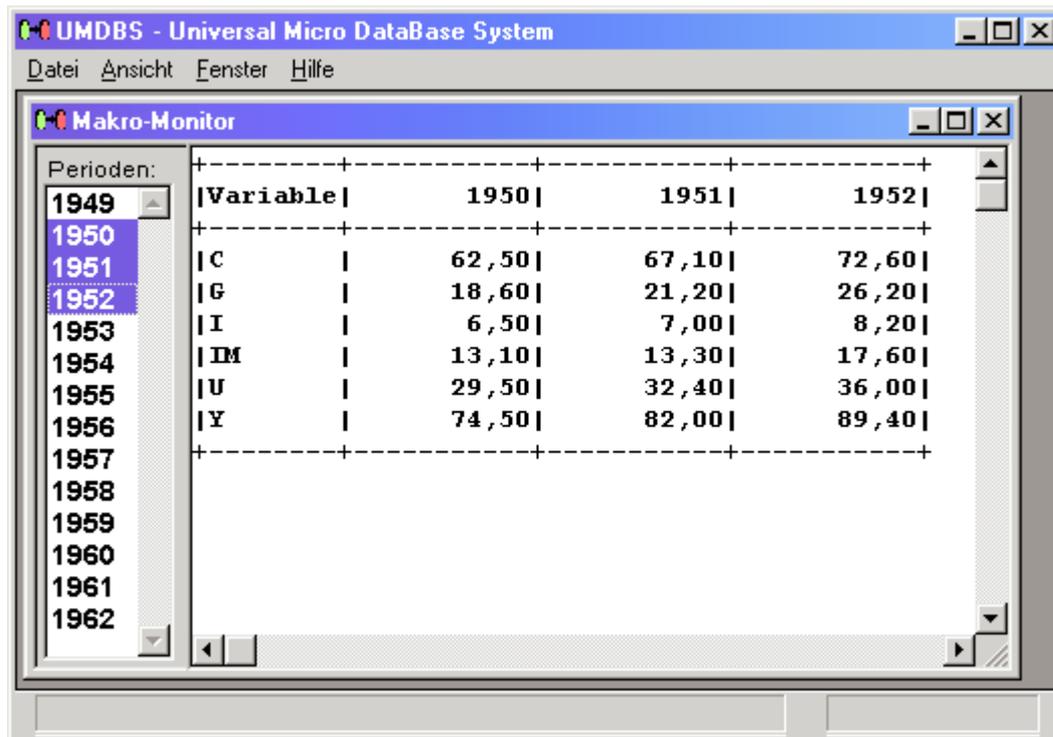


Links oben sind die verfügbaren Variablenarten aufgelistet. Bei der Simulation werden unter der Bezeichnung "endogen" die Interface-Variablen vom Typ OUTPUT aufgelistet, unter "Ereignisse" die über den MISTRAL-Befehl "EVENT" erfaßten (gewichteten) Häufigkeiten. Darunter können die innerhalb einer Variablenart verfügbaren Zeitreihen ausgewählt werden. Durch einen Doppelklick werden die Zeitreihen in die Darstellung aufgenommen oder daraus entfernt. Ein Zugriff auf Befehle bereits angezeigter Variablen besteht über das Kontextmenü (rechte Maustaste) der Variablennamen in der Grafiklegende.

Die letzten beiden hinzugefügten Variablen (in der Legende mit rotem Strich gekennzeichnet) können wahlweise auch in einem Streudiagramm dargestellt werden.

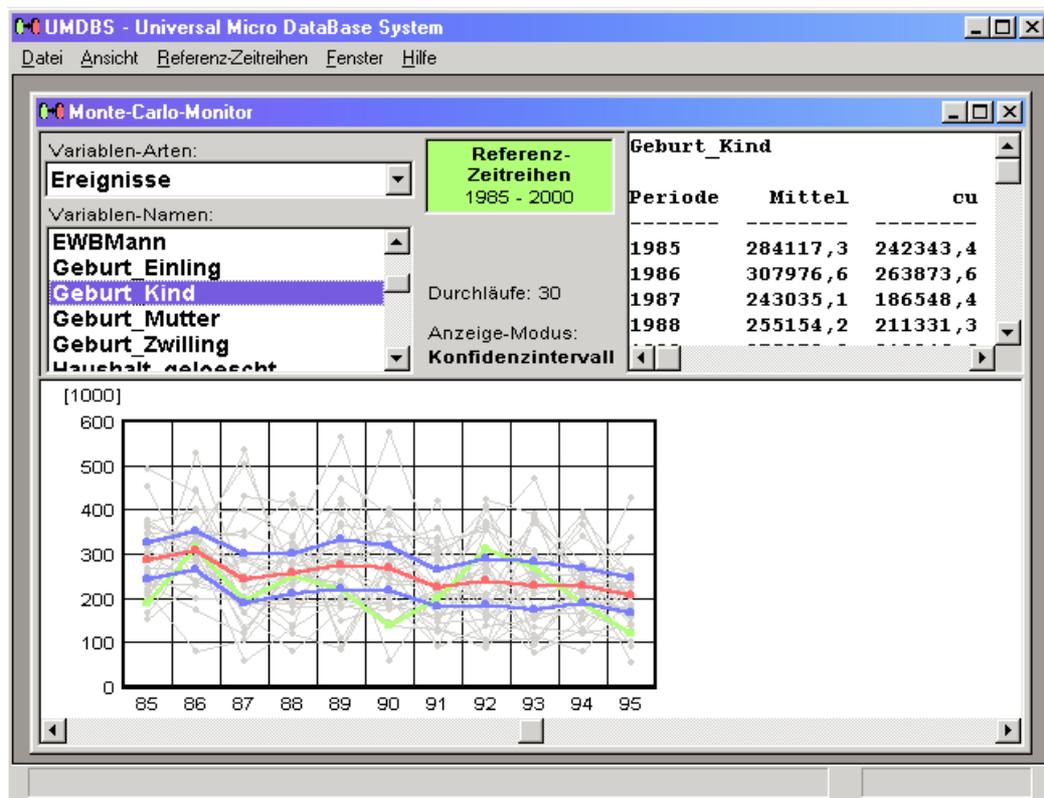
## 5.5 Makro-Monitor

Der Makro-Monitor dient ebenso wie der Zeitreihen-Monitor zur Darstellung von Daten aus einer Zeitreihendatenbasis. Sein Schwerpunkt liegt jedoch auf der schnellen, ausschließlich tabellarischen Anzeige aller Variablen einer oder mehrerer Perioden:



## 5.6 Monte-Carlo-Monitor

Der Monte-Carlo-Monitor dient dazu, die Ergebnisse einer Monte-Carlo-Simulation darzustellen und zu analysieren. Er ähnelt in gewisser Weise dem Zeitreihen-Monitor.



Links oben können auch hier die Variablenart und die Variable (hier nur eine!) ausgewählt werden. In der Grafik werden (optional) alle durchgeführten Simulationsläufe gleichzeitig dargestellt. Die eingeblendeten farbigen Markierungen kennzeichnen je nach Auswahl entweder das Konfidenzintervall für den Mittelwert der stochastischen Simulation, den arithmetischen Mittelwert sowie Mittelwert  $\pm 1$ - bzw. 2-mal Standardabweichung für die konkreten Simulationsläufe oder die Quartile inkl. Minimum und Maximum dafür.

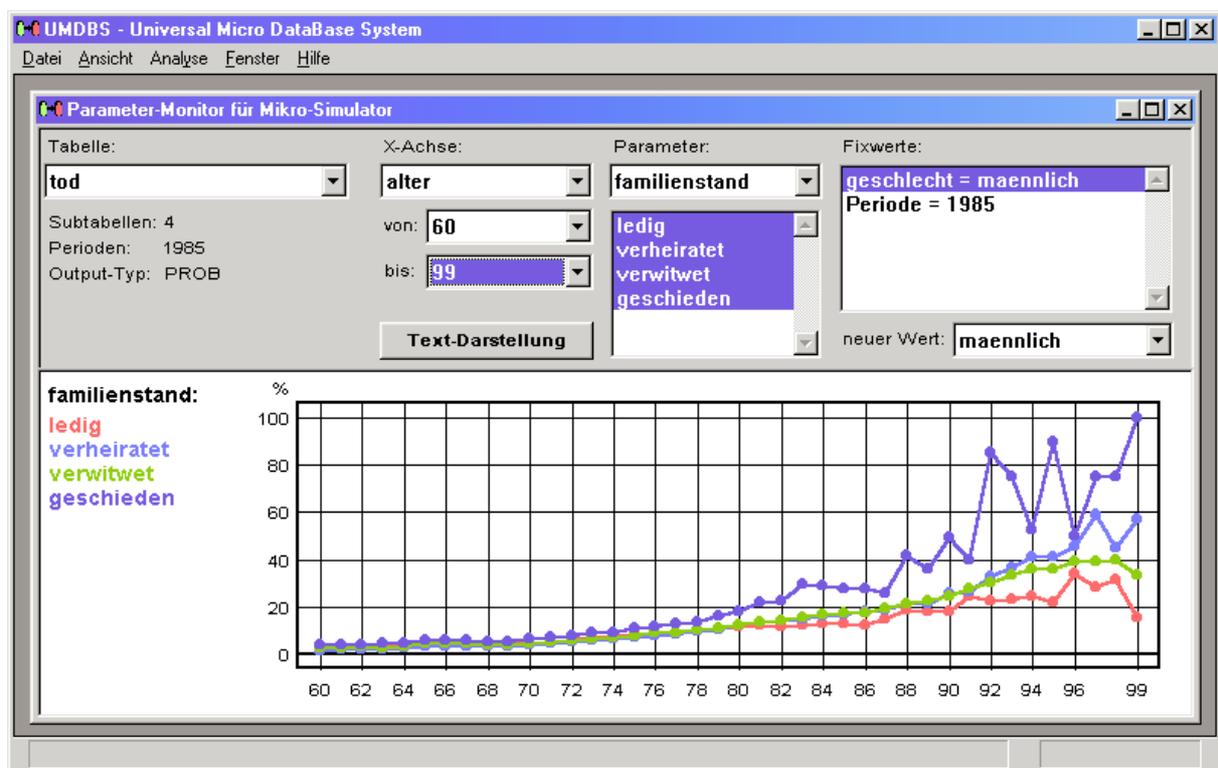
Rechts oben stehen diese und weitergehende Informationen als Tabelle zur Verfügung. Zusätzlich können weitere Informationen in benutzerkonfigurierbaren Reports ausgegeben werden.

Zum automatischen Vergleich mit Referenz-Zeitreihen (z.B. bei der Ex-Post-Simulation die echten Werte) kann - wie in den Flußdiagrammen - eine Zeitreihendatenbasis geladen werden. Sofern sich dort eine Zeitreihe mit demselben Namen wie die dargestellte befindet, werden deren Werte in das Diagramm eingeblendet sowie in die Tabelle und die Reports aufgenommen.

Zusätzlich können auch direkt der Zeitreihen-Monitor und der Makro-Monitor aufgerufen werden. Diese erhalten dann eine Zeitreihendatenbasis mit den Mittelwerten aller Simulationsläufe.

## 5.7 Parameter-Monitor

Der Parameter-Monitor dient dazu, die im Parameter-Modul als Text eingegebenen Daten grafisch darzustellen:



Links oben wird die gewünschte Tabelle ausgewählt. Rechts daneben wird ein Merkmal für die X-Achse bestimmt. Sofern mehr als ein Wert bzw. eine Wertereihe angezeigt werden soll, kann unter "Parameter" ein Merkmal ausgewählt werden. Von diesem können dann beliebig

viele Ausprägungen selektiert werden, zu denen jeweils eine eigene Darstellung erfolgt. Die übrigen Argumente der Tabelle müssen auf einen festen Wert eingestellt werden, der rechts oben eingestellt werden kann.

Bei Anklicken der Schaltfläche "Text-Darstellung" werden die Daten in einem getrennten Fenster in Tabellenform angezeigt, so daß z.B. genaue Werte abgelesen werden können.

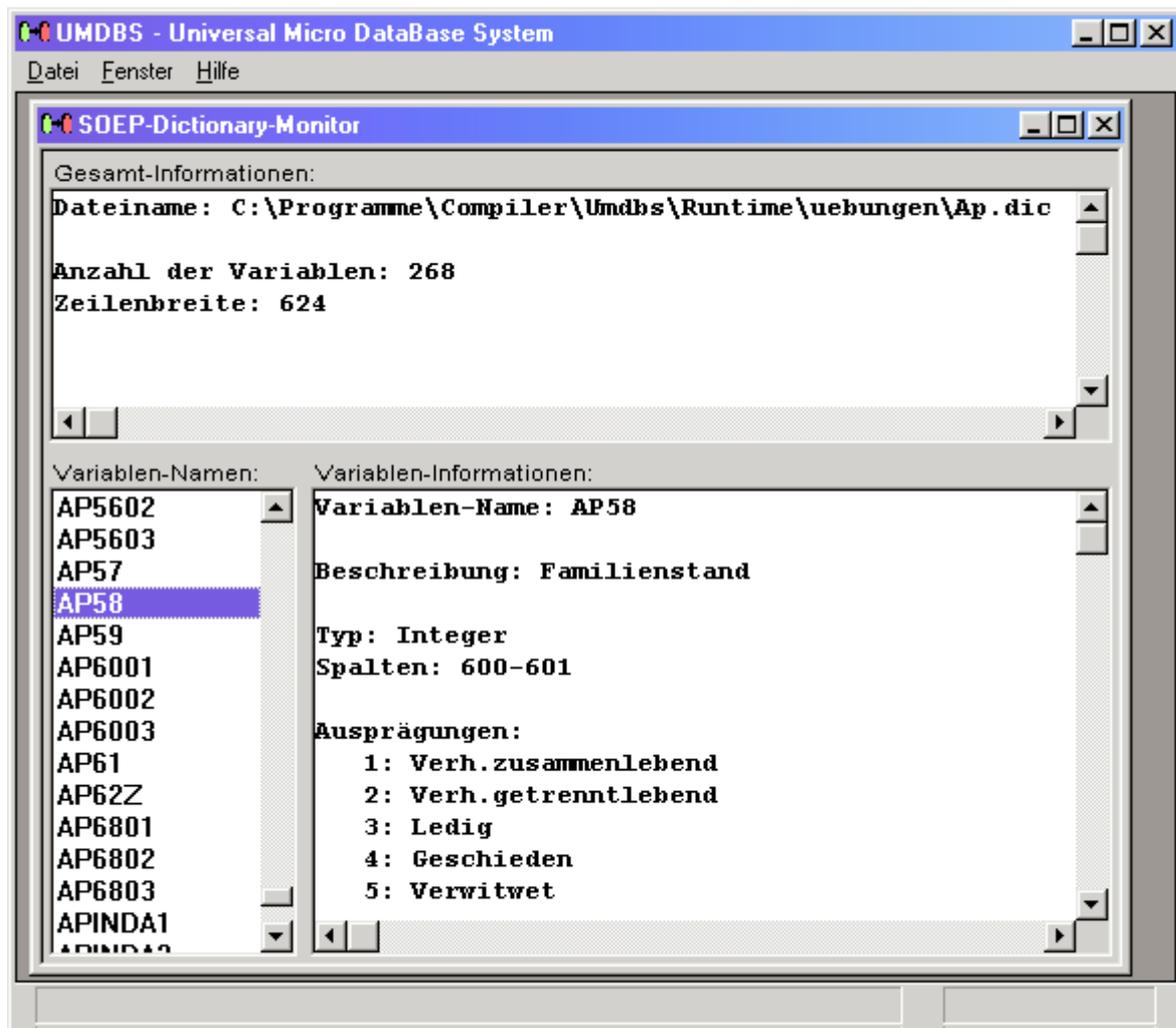
Der Parameter-Monitor geht jeweils davon aus, daß für einzelne (bzw. bei X-Achse und Parametern mehrere) Ausprägungen der Argumente jeweils genau ein Wert (oft eine Wahrscheinlichkeit) existiert. Bei einigen Tabellen wird hingegen ein Zufallswert als Input übergeben und gemäß den in der Tabelle definierten Wahrscheinlichkeiten einer von mehreren möglichen Ergebniswerten zurückgegeben. Um auch diese Art von Tabellen visualisieren zu können, wird dafür die Richtung der Daten umgekehrt. Dazu werden die möglichen Ergebnisausprägungen wie ein Eingangs-Argument behandelt, während die eigentlich als Input vorgesehene Zufallszahl durch Wahrscheinlichkeiten repräsentiert wird, die in der Darstellung als Ergebnis erscheinen. Diese Anzeigeform wird unterhalb des Tabellennamens links oben mit dem Hinweis "Inverser Modus" verdeutlicht.

## 6 Hilfsprogramme

Hilfsprogramme innerhalb des UMDBS dienen dazu, Zusatzfunktionen zu realisieren, die insbesondere im Zusammenhang mit externen Systemen stehen.

### 6.1 SOEP-Dictionary-Monitor

Der SOEP-Dictionary-Monitor dient der Analyse der \*.dic-Dateien des SOEP, die Metainformationen zu den eigentlichen Datenfiles (\*.dat) beinhalten. Beim Aufruf ist der Name des \*.dic-Files anzugeben. Danach öffnet sich folgendes Fenster:



Im oberen Bereich werden Informationen zur gesamten Datei angezeigt. Links unten sind alle verfügbaren Variablen der Datei alphabetisch aufgelistet und können (immer nur eine) durch Anklicken ausgewählt werden. Alternativ kann über eine Suchfunktion eine Variable gesucht werden, die einen Suchbegriff (inkl. Wildcards) innerhalb des Namens, der Beschreibung oder in den Ausprägungen trägt. Die Angaben zur jeweils gewählten Variable werden rechts unten dargestellt.

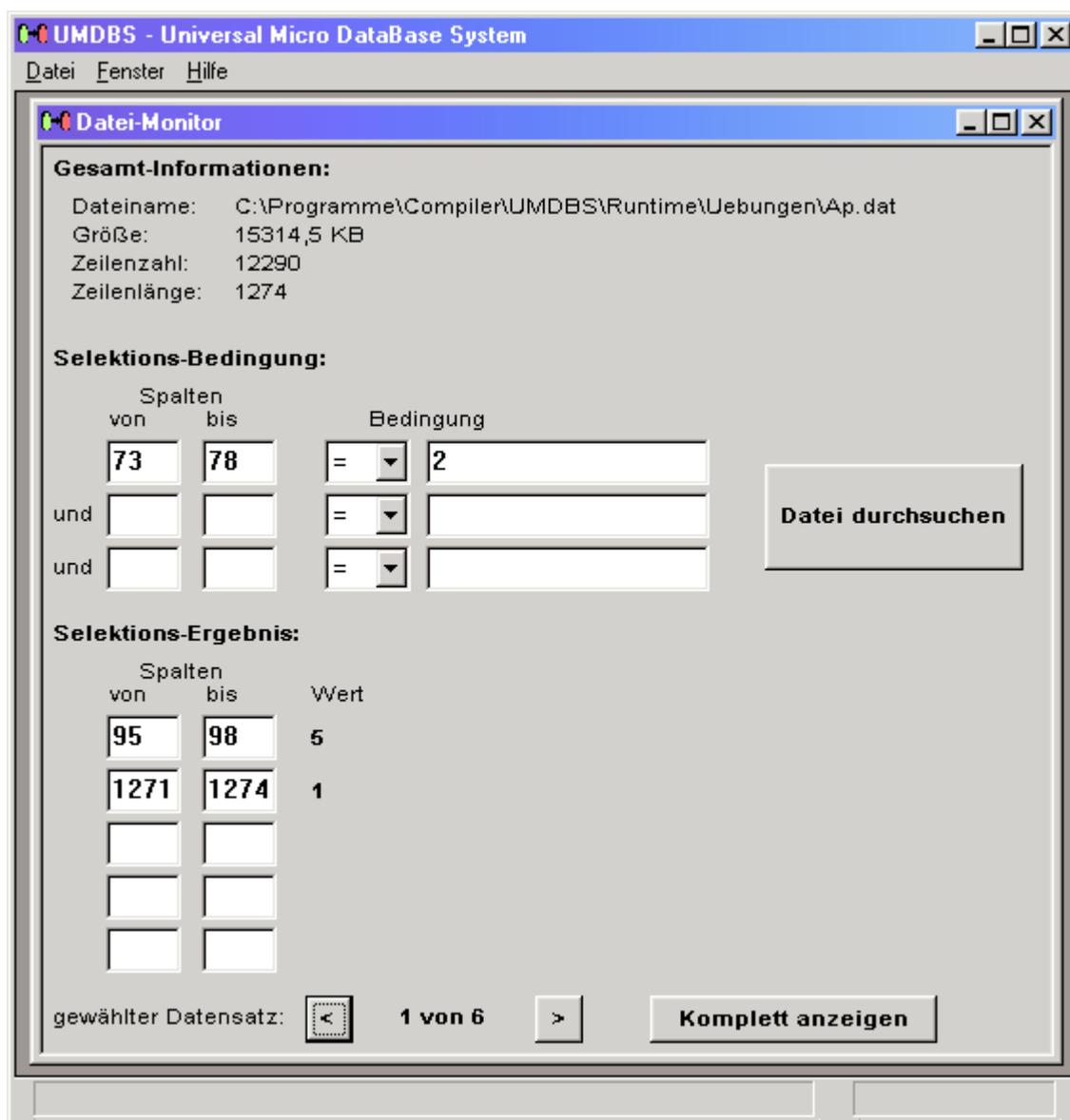
Für das Einlesen innerhalb eines Generierungsprogramms ist die Angabe der genauen Position (Spalten) sowie die Kodierung der Ausprägungen entscheidend. Zur Erleichterung kann über

einen Befehl (über Popup-Menü im rechten unteren Fenster) automatisch MISTRAL-Code generiert werden, der das Erstellen von Import-Routinen wesentlich vereinfacht und Fehler vermeiden hilft.

Eine zweite nützliche Funktion ist die automatische Untersuchung der dazugehörigen \*.dat-Datei für das ausgewählte Merkmal. Damit werden die Häufigkeiten angezeigt, mit der die einzelnen Ausprägungen bei den Mikroobjekten innerhalb des SOEP vorkommen. Dadurch können ohne Aufwand wertvolle Vorabinformationen gewonnen werden.

## 6.2 Datei-Monitor

Der Datei-Monitor dient dazu, strukturierte ASCII-Dateien, insbesondere solche mit Mikrodaten, zu untersuchen.



Das Fenster ist im wesentlichen in drei Bereiche untergliedert:

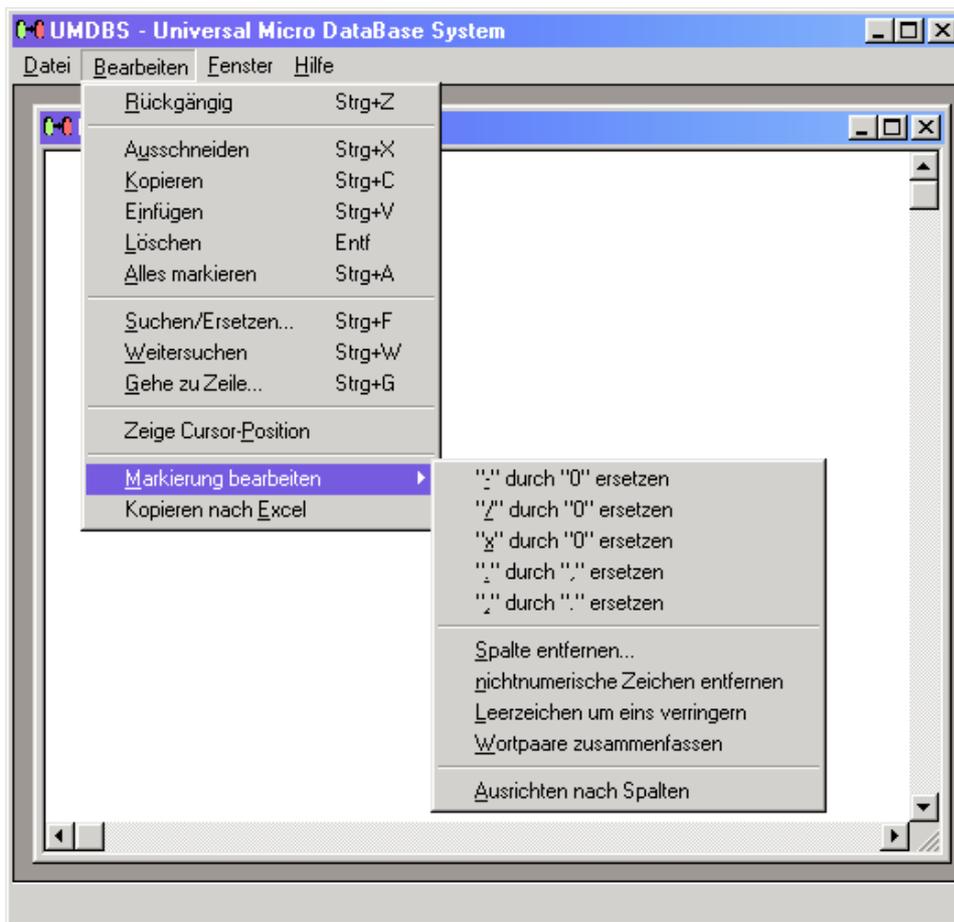
Oben befinden sich allgemeine Informationen zu der geöffneten Datei.

In der Mitte besteht die Möglichkeit, Datensätze nach einer Selektionsbedingung auszuwählen. Dabei wird vorausgesetzt, daß sich die wesentlichen Informationen zu jedem Datensatz in jeweils einer eigenen Textzeile befinden und in festen Spalten angeordnet sind. Der Benutzer kann bis zu drei Bedingungen festlegen, die sich auf den Wert bestimmter Spalten beziehen. Dazu ist zu jedem Attribut die erste und die letzte Spalte anzugeben (beginnend mit 1). Das so festgelegte Attribut wird mit einem festen Wert verglichen, wobei alle sechs üblichen Vergleichs-Operatoren zur Auswahl stehen. Werden mehrere Bedingungen angegeben, müssen alle erfüllt sein, damit ein Datensatz ausgewählt wird. Gestartet wird die Selektion der Datensätze mit dem Button "Datei durchsuchen".

Im unteren Bereich können bis zu fünf Attribute über ihre Spaltenpositionen festgelegt werden, die für die zuvor selektierten Datensätze angezeigt werden. Zwischen den Datensätzen, deren Anzahl unten angezeigt wird, kann mit den beiden Pfeil-Schaltflächen umgeschaltet werden. Mit dem Button "Komplett anzeigen" wird die gesamte Textzeile des aktuellen Datensatzes in einem zusätzlichen Textfenster (zusammen mit Markierungen mit den Spaltenpositionen) angezeigt.

### 6.3 Import-Editor

Der Import-Editor dient vor allem dazu, Tabellen, die über ein Texterkennungssystem (OCR) eingescannt wurden, aufzubereiten. Die wesentlichen Teile der Funktionalität befinden sich im Menü "Bearbeiten".



Da in der Regel verschiedene Teile des Textes unterschiedlich zu bearbeiten sind, beziehen sich die Befehle des Untermenüs "Markierung bearbeiten" jeweils nur auf den zuvor markierten Bereich. Hier eine Kurzbeschreibung der verfügbaren Kommandos:

- Die fünf obersten Kommandos dienen dazu, Umwandlungen für die Weiterverarbeitung vorzunehmen, die sich einerseits auf die in statistischen Tabellen üblichen Schreibweisen beziehen und andererseits zur Konvertierung zwischen Dezimalpunkt und -komma dienen.
- Mit "Spalte entfernen..." kann eine bestimmte Spalte (Spalten-Nr. wird vom Benutzer abgefragt) vollständig gelöscht werden. Da jedes Wort (ohne Leerzeichen) eine eigene Spalte darstellt, ist zuvor auf eine entsprechende Bearbeitung zu achten (z.B. auch durch das Scannen hinzugefügte Punkte o.ä. entfernen!).
- "nichtnumerische Zeichen entfernen" dient vor allem dazu, Überschriften o.ä. zu löschen. Zuvor ist sicherzustellen, daß sich in den Zahlen keine solchen Zeichen mehr befinden, die auf Fehlerkennungen basieren.
- "Leerzeichen um eins verringern" kann vor allem dazu genutzt werden, um den üblichen Abstand zwischen den 1000er-Stellen zu eliminieren, da diese sonst als zwei getrennte Zahlen aufgefaßt werden. Dazu wird aus einer Folge von Leerzeichen jeweils eins gelöscht. Einzelne Leerzeichen verschwinden völlig, so daß eng zusammenstehende Wörter bzw. Zahlen zu einem Wort werden.
- "Wortpaare zusammenfassen" ist eine alternative Möglichkeit, zusammengehörige Teile einer Zahl (Trennung an der 1000er-Stelle) zusammenzufassen. Dabei werden, beginnend mit jeder Zeile, die Wörter eins und zwei zu einem Wort, ebenso die Wörter drei und vier usw. Dieser Befehl ist deshalb nur dann zu nutzen, wenn alle Zahlen des markierten Bereichs aus genau zwei Teilen bestehen.
- Mit "Ausrichten nach Spalten" werden alle Wörter einer Spalte korrekt untereinander rechtsbündig ausgerichtet. Dies hilft auch dabei, kleine "Schmutzeffekte" (z.B. eingescannte Punkte oder Kommas) sicher zu erkennen.

Daneben stehen natürlich auch alle sonst üblichen Bearbeitungsoperationen (inkl. Suchen/Ersetzen) zur Verfügung.

Der Import-Editor ist für die Zusammenarbeit mit Excel optimiert. Um einen Bereich dorthin zu übertragen, wird er zunächst mit der Maus markiert, mit "Kopieren nach Excel" in die Zwischenablage gestellt und in Excel mit "Einfügen" in einen freien Tabellenbereich kopiert. Beim Markieren ist darauf zu achten, daß keine überflüssigen Leerzeilen mit markiert werden, da diese ebenfalls nach Excel übertragen werden.

# Anhang

## A.1 Installation

Das UMDBS besteht aus ca. 90 Dateien der Typen \*.dll und \*.sll (Smalltalk Link Libraries) sowie einer einzigen Programmdatei umdbs.exe, die sich alle innerhalb eines beliebigen Verzeichnisses befinden müssen (z.B. auch auf einer CD-ROM). Das Programm kann einfach durch Aufrufen der Datei umdbs.exe (z.B. durch Doppelklick im Windows-Explorer) gestartet werden.

Eine Installation im umfassenden Sinne (z.B. inkl. Einträgen in die Registry oder andere Systemdateien und -verzeichnisse) ist nicht notwendig. Zum Einrichten auf einem Rechner müssen nur die Dateien in folgender Weise auf die lokale Festplatte (oder auch ein Remote-Laufwerk) kopiert werden:

1. Anlegen eines neuen Verzeichnisses (z.B. ..\umdbs).
2. Kopieren der Dateien umdbs.exe, \*.dll und \*.sll in dieses Verzeichnis.
3. Es wird empfohlen, ein Verzeichnis "daten" o.ä. unterhalb des Installationsverzeichnisses anzulegen, das die Daten und MISTRAL-Programme aufnimmt.
4. Optional kann manuell ein entsprechender Eintrag in das Start-Menü von Windows oder ein Symbol auf dem Desktop angelegt werden, das mit der Datei umdbs.exe verknüpft ist.

Die aktuelle Distribution des UMDBS läßt sich auf nur vier Disketten mit komprimierten Dateien unterbringen. Verfügbar sind alle Dateien über das Internet unter dem URL

<http://mitarbeiter.fh-heilbronn.de/sauerbier/umdbs>

Für den Download ist ein Benutzername und Paßwort notwendig, die beim Autor erhältlich sind. Folgende ZIP-Dateien sind vorhanden:

Programm:

umdbs.zip	Programm umdbs.exe sowie readme.txt mit Änderungsinformationen
umdbs_ <i>sn</i> .zip	drei Dateien ( <i>n</i> steht für 1, 2 und 3) mit unveränderlichem Grundsystem aus *.dll und *.sll; Aufteilung auf drei Dateien, die jeweils auf eine Diskette passen

Dokumentation:

doku.zip	MISTRAL- und UMDBS-Handbuch als PDF-Dateien
tutorial.zip	integriertes Übungshandbuch (HLP-Datei) inkl. Beispieldateien, die automatisch in das Unterverzeichnis "uebungen" entpackt werden

Updates im Rahmen der Produktpflege und -erweiterung bestehen jeweils nur aus einer neuen Datei umdbs.zip.

Um da UMDBS vollständig nutzen zu können, ist eine **Lizenzdatei** namens "reginfo.ini" erforderlich, die interessierten Benutzern kostenlos vom Autor Prof. Dr. Thomas Sauerbier zur Verfügung gestellt wird (Email an sauerbier@fh-heilbronn.de). Ohne diese Lizenz ist die Simulation bis maximal 5.000 Mikroobjekte möglich (keine sonstigen Einschränkungen).

Das UMDBS ist auf jedem System lauffähig, auf dem Windows 95, 98, NT oder 2000 läuft. Der Speicherbedarf auf der Festplatte liegt mit der kompletten Dokumentation bei etwa 15 MB; der RAM-Bedarf im Betrieb hängt hauptsächlich von der Menge der jeweils verarbeiteten Mikrodaten ab (mindestens 64 MB empfohlen).

## **A.2 Lizenzrechtliche Hinweise**

Das UMDBS ist mit der Programmierumgebung Visual Smalltalk von ParcPlace Digitalk entwickelt worden. Das System wird gemäß den Lizenzbedingungen von Visual Smalltalk nur mit Runtime-Dateien ausgeliefert, die keine Komponenten der Entwicklungsumgebung beinhalten. Die Rechte am UMDBS liegen beim Autor Prof. Dr. Thomas Sauerbier.

Das UMDBS darf von berechtigten Nutzern im Rahmen ihrer Forschungsarbeiten verwendet werden. Dazu wird eine Lizenzdatei erstellt, die auf den jeweiligen Nutzer bzw. das entsprechende Forschungsinstitut lautet. Eine kommerzielle Nutzung oder Weitergabe an Dritte bedarf der ausdrücklichen Genehmigung des Autors.

### A.3 Grenzen der aktuellen Implementierung

Das UMDBS ist ein System, dessen einfache und schnelle Handhabung unter anderem darauf basiert, daß die bearbeiteten Daten unmittelbar im Hauptspeicher (RAM) gehalten werden. Entsprechend ist ein großer Arbeitsspeicher sinnvoll bzw. notwendig, um mit größeren Datenmengen zu arbeiten. Trotzdem sind die Grenzen des aktuellen UMDBS relativ weit gezogen:

- Bezüglich der **Anzahl der Mikroobjekte** ist bisher keine Beschränkung festgestellt worden; auch mehr als 500.000 Objekte konnten importiert werden. Allerdings limitiert der verfügbare bzw. vom Simulator verwaltbare Hauptspeicher die Anzahl der Objekte in Abhängigkeit von der Anzahl und Art ihrer Attribute (siehe dazu auch das MISTRAL-Handbuch). Mit 64 MB RAM sollten auch umfangreiche Modelle auf der Basis des Sozioökonomischen Panels (SOEP) mit ca. 6.000 Haushalte verarbeitet werden können; 96 MB reichen nach aktuellem Kenntnisstand für alle Modelle auf Basis der Einkommens- und Verbrauchsstichprobe (EVS) mit ca. 50.000 Haushalte aus. Eine Vergrößerung des Hauptspeichers über 96 MB hinaus bringen keine Verbesserung. Die Daten des gesamten Mikrozensus können hingegen nicht als eine Mikrodatenbasis mit dem UMDBS verarbeitet werden. Im Mikroobjekt-Monitor und Mikroobjekt-Editor können in den Listen mit den Objekt-Schlüsselnummern nicht mehr als 32.766 Mikroobjekte gehalten werden. Mit Hilfe von MQL-Ausdrücken können bei Bedarf aber kleinere Untermengen gebildet werden (u.a. auch anhand der Schlüsselnummern), die sich dann wieder ohne Einschränkungen bearbeiten lassen.
- **Textdateien für den Import** können beliebig groß sein. Erfolgreich eingelesen wurde bisher Dateien mit bis zu 180 MB Größe.
- Der **erzeugte Report** stößt ab ca. 20 MB an eine Grenze. Dies dürfte jedoch nur bei extrem umfangreichen Trace-Dateien erreicht werden.

## A.4 Änderungen zu früheren Versionen des UMDBS

### Änderungen der Version 3.0 gegenüber Version 2.0

#### **Generell:**

- UMDBS ist jetzt zweisprachig (deutsch / englisch umschaltbar).
- In der Datei "umdb.ini" werden Benutzereinstellungen (z.Z. gewählte Sprache und Default-Verzeichnis für Benutzerdateien) abgelegt.
- Zur uneingeschränkten Benutzung (Simulation von mehr als 2.000 Mikroobjekten) ist eine Lizenzdatei erforderlich, die für interessierte Benutzer kostenlos erhältlich ist.
- Die Dokumentation wurde erweitert und steht jetzt direkt aus dem UMDBS zur Verfügung.
- Viele Verbesserungen und Erweiterungen im Detail.
- An vielen Stellen wurde die Oberfläche mehr oder weniger stark überarbeitet.
- Verschiedene Erweiterungen in MISTRAL (siehe dazu entsprechende Handbuch).

#### **Neue Werkzeuge:**

- Der Datei-Monitor erlaubt die Analyse von Textdateien (insb. mit Mikrodaten o.ä.).
- Der Import-Editor dient vor allem dazu, per Texterkennung eingescannte Tabellen (insb. im Zusammenhang mit Excel) aufzubereiten und zu korrigieren.

#### **Wichtige Ergänzungen vorhandener Werkzeuge:**

- Vollständig überarbeiteter Monte-Carlo-Monitor mit umfangreichen, konfigurierbaren Reports, der Ausgabe von Konfidenzintervallen und dem automatischen Vergleich mit Referenz-Zeitreihen.
- Deutlich verbessertes Arbeiten mit Include-Dateien, u.a.
  - Zusammenfassen aller Includedateien zu einer einzigen Datei
  - direkter Aufruf einer untergeordneten Include-Datei
  - Anzeige der Include-Hierarchie in den Info-Texten
- Verbesserte Schnittstellen zu Excel (Kopieren von Daten in beide Richtungen).
- Komfortabler Daten-Import mit automatischer Programm-Generierung für den Zeitreihen-Editor.
- SOEP-Dictionary-Monitor mit Gesamtreport zu einer \*.dic-Datei.

#### **Sonstiges:**

- Das Hilfsprogramm DPMS-Parameter-Import wurde - da nicht mehr benötigt - entfernt.

## Index

### A

Analyse.....	16
Analysemodell.....	12
Analysemodul .....	15, 16
Arbeitsspeicher.....	37
ASCII-Datei .....	31
ASCII-Text .....	19
Attribute eines Mikroobjekts.....	21, 23, 24

### B

Benutzeinstellungen .....	7
---------------------------	---

### C

CD-ROM.....	35
Compiler .....	19

### D

Dateien .....	6, 8
Datei-Monitor .....	31
Dateitypen .....	6
Default-Pfad für die Benutzdateien.....	7
DISTRIBUTION .....	11
DISTRIBUTION .....	14
Dokumentation.....	35
Durchlauf .....	16

### E

Editoren.....	19
textbasierend .....	19
Einkommens- und Verbrauchsstichprobe.....	37
EVENT .....	26
EVS.....	37
Excel .....	20, 33

### F

Flußdiagramme .....	9, 10
Fortschreibungsalgorithmus .....	14

### G

Generierung einer Mikrodatenbasis .....	17
Generierungsfunktionen .....	20
Generierungsmodell .....	12, 18
Generierungsprogramm.....	30

### H

Hauptspeicher .....	37
Hilfsprogramme .....	30
Histogramm.....	25

### I

Import einer Mikrodatenbasis .....	17
Importdateien .....	13
Import-Editor .....	32
Info-Text .....	20
Initialisieren .....	20
von Modulen .....	10

Installation .....	35
Interface-Variable .....	11
Interface-Variablen .....	26
Internet .....	35
Inverser Modus .....	29

### K

Key.....	21
Klasse eines Mikroobjekts .....	21
Kreisdiagramm.....	25
Kurz-Präsentation des Systems .....	8

### L

Lizenzdatei.....	35, 36
Lizenzrechtliche Hinweise .....	36

### M

Makro-Monitor .....	23, 26
MDI-Oberfläche.....	5
Mergen .....	13
Mikrodatenbasis.....	7, 10, 14, 15, 16, 17, 21, 23, 24
Generierung.....	12
Objekte löschen.....	22
Mikro-Monitor .....	23, 24, 25
Mikroobjekte, maximale Anzahl.....	37
Mikroobjekt-Editor .....	21
Mikroobjekt-Monitor .....	21, 23
MISSING_VALUES .....	13
MISTRAL.....	10
MISTRAL-Handbuch .....	8
MISTRAL-Programmcode .....	19
MML-Modell .....	19
Module .....	10
Monitore .....	23
Monte-Carlo-Ergebnisse .....	23
Monte-Carlo-Monitor .....	15, 23, 27
Monte-Carlo-Simulation .....	15, 27
Monte-Carlo-Streuung .....	15
SQL .....	11, 14, 21, 24, 25

### N

NIL.....	22
----------	----

### O

OCR .....	32
Online-Dokumentation.....	8
OUTPUT .....	26

### P

Parameterdatenbasis.....	7, 11, 13, 14, 19, 23
Parameter-Editor .....	20
Parameter-Modul .....	28
Parameter-Monitor.....	23, 28
PDF-Dateien .....	8
Programmcode .....	19
Programmversion .....	8

## Index

---

### **R**

RAM .....	37
Readme-Datei.....	8
reginfo.ini.....	7, 35
Registry .....	35
Report.....	11
maximale Größe .....	37
Reportdatei.....	7

### **S**

Säulendiagramm.....	25
Schlüsselnummer eines Mikroobjekts .....	21, 24, 37
Seed.....	21
SET .....	22
Simulation .....	13
erweiterte.....	14
Monte-Carlo .....	15
Standard .....	13
stochastische.....	15
Simulationsmodell.....	10, 12
Smalltalk Link Libraries.....	35
SOEP.....	7, 30, 37
SOEP-Daten-File .....	7
SOEP-Dictionary-File.....	7
SOEP-Dictionary-Monitor .....	30
Stapelsäulendiagramm .....	25
Streudiagramm .....	25
Syntax-Fehler .....	19

### **T**

Testmodell .....	12
Testmodul .....	15, 16
Textbearbeitung .....	20
Textdateien .....	18
maximale Größe .....	37
Texterkennungssystem.....	32
Trace-Datei .....	37

### **Ü**

Übungs-Handbuch .....	8
umdbbs.ini.....	7
UMDBS-Handbuch.....	8

### **V**

Verteilungsdatenbasis .....	7, 11, 13, 14, 23
Verteilungs-Monitor.....	23, 25

### **W**

WRITELN .....	12
---------------	----

### **Z**

Zeitreihendatenbasis.....	7, 11, 19, 23, 26
Zeitreihen-Editor.....	20
Zeitreihen-Monitor.....	23, 26
ZIP-Dateien.....	35
Zufallsgenerator .....	21
Zufallsstreuung.....	15