

Realisierung der grafischen Modellierungsmethode Dynamic Relations für zeitreihenbasierte Modelle mit EXTEND

Dietmar Klinger, Bernd Page und Volker Wohlgemuth¹

Abstract

Dieser Beitrag beschreibt die Implementation einer grafischen Modellierungsmethode für zeitreihenbasierte Modelle für die Verkehrsemissionsmodellierung mit dem Simulationswerkzeug EXTEND. Es wird zunächst der Modellierungsansatz vorgestellt. Im Anschluß hieran skizzieren wir die grundsätzlichen Merkmale des offenen Simulationssystems EXTEND. Abschließend wird die Implementation der beschriebenen Modellierungsmethodik mit EXTEND beschrieben und die Vorteile der Verwendung eines offenen Simulationswerkzeuges in diesem Anwendungsbereich aufgezählt.

1 Motivation

Seit längerer Zeit befaßt sich die Arbeitsgruppe von Prof. Dr. Bernd Page mit der informationstechnischen Unterstützung der Modellbildung und Simulation im Umweltbereich. Dabei steht insbesondere die Frage der Eignung bestehender Simulationswerkzeuge im jeweiligen Anwendungskontext im Vordergrund der Untersuchung. In diesem Zusammenhang taucht nach der Anforderungsphase immer wieder die Frage auf, ob die festgestellten, spezifischen Anforderungen des jeweiligen Anwendungsgebietes nur mit Hilfe einer Neuentwicklung eines spezifischen Simulationswerkzeuges umzusetzen sind, oder ob es nicht effektiver und schneller ist, auf der Basis eines allgemeinen Simulationswerkzeuges eine spezielle Erweiterung für das zu untersuchende Anwendungsgebiet zu implementieren. Dabei handelt es sich letztendlich auch um die in anderen Kontexten der Informatik häufig gestellte Frage, ob Individualsoftware der Standardsoftware vorzuziehen ist. Aus diesem Grunde wurde in mehreren Arbeiten versucht, das allgemeine, offene Simulationswerkzeug EXTEND um spezielle Komponenten für verschiedenste Anwendungsgebiete im Umweltbereich (z.B. Expositionsanalyse von Chemikalien in der Umwelt, Räuber-Beute-Modelle, Umsetzung der System Dynamics-Methodik) zu erweitern. Dabei zeigte sich, daß EXTEND aufgrund seiner offenen Architektur hierfür recht gut ge-

¹ Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Straße 30, D-22527 Hamburg,
email: {2klinger|page|wohlgemuth}@informatik.uni-hamburg.de

eignet erscheint (Grünewald 1997, Page et al. 1997, Theilmann/Wolter 1999, Wohlgemuth 1997).

Als weiteren Baustein dieser Untersuchungsreihe stellt dieser Beitrag den Anwendungsbereich der zeitreihenbasierten Verkehrsträgeremissionsmodellierung vor und beschreibt in diesem Zusammenhang die schon früher von der Arbeitsgruppe entwickelte Methodik Dynamic Relations (Freese/Seidel 1994, Fresse et al. 1994).

2 Die Modellierungsmethodik Dynamic Relations

Die Methodik Dynamic Relations entstand am Fachbereich Informatik der Universität Hamburg im Rahmen eines geförderten Kooperationsvorhaben mit dem Umweltbundesamt. Im Umweltbundesamt sammeln Verkehrsexperten unter anderem Informationen über die Schadstoffemissionen im Autoverkehr der Bundesrepublik Deutschland. Diese Informationen werden zur Analyse vergangener Emissionsentwicklungen benutzt, wobei die daraus gewonnenen Erkenntnisse Prognosen zukünftiger Emissionsbelastungen ermöglichen sollen. Von großem Interesse ist dabei die Auswirkung von Schadstoffminimierungstechnologien auf die Emissionsentwicklung. Zu diesem Zwecke werden im Umweltbundesamt analytische Modelle zur Berechnung der Schadstoffemissionen erstellt. Diese Modelle basieren auf empirischen verkehrsbezogenen Daten (z.B. Zulassungszahlen pro Jahr, mittlere Fahrleistung pro Jahr nach Fahrzeugtyp und Alter, Emissionsfaktoren nach Schadstoffen, etc.), welche in Tabellenform vorliegen. Hauptbestandteil der Emissionsuntersuchung ist, durch Verknüpfung verschiedener Tabellen neue Informationen zu gewinnen (Niederle 1991).

Ziel der Entwicklung war es daher, eine Methodik in einem Softwarewerkzeug darzustellen, mit dem auch Benutzer ohne große DV-Kenntnisse tabellenbasierte Modelle eigenständig und ohne großen Arbeitsaufwand erstellen, simulieren und auswerten können. Dabei soll die Modellerstellung durch den Aufbau eines grafischen Modelldiagramms vereinfacht und eine höhere Modelltransparenz geschafft werden. Das Modelldiagramm soll außerdem die Modellgrößen und ihre Beziehungen auf anschauliche Weise darstellen und Benutzern, die den Prozeß der Modellbildung nicht verfolgt haben, eine leichte Einarbeitung ermöglichen. Der Benutzer soll die Simulationsläufe interaktiv gestalten und variieren können. Außerdem war eine grafische Aufarbeitung wichtiger Tabellendaten in Kurvenform erwünscht.

Die Methode Dynamic Relations baut auf dem relationalen Datenmodell auf, wie es von E. F. Codd 1970 definiert wurde. Hierbei dient die Tabellenform zur Darstellung einer Relation: Die Spaltennamen der Tabelle entsprechen den Attributnamen der Relation, die Feldwerte stehen für die Attributwerte und die Zeilen symbolisieren die Tupel. In der Definition von E. F. Codd muß ein minimales relationales Datenmodell die Operationen Selektion, Projektion und Kombination von Tabellendaten unterstützen. Dieses Minimalmodell reicht aber bei weitem nicht aus, die oben

genannten Anforderungen zu erfüllen. Es wurde deshalb die Methodik Dynamic Relations um Operationen zur arithmetischen Verknüpfung, zur Gruppenbildung und Datenverdichtung und zur Prognose erweitert. Bei sämtlichen Operationen werden Tabellen verarbeitet und neue Tabellen erstellt. Diese Einheitlichkeit ermöglicht die Kombination von Operationen und somit die Ausführung komplexer Operationen.

Dynamic Relations erlaubt somit die Berechnung von in den Tabellen nicht bestimmter Werte sowie die Erstellung von Prognosen. Implementiert wurde die Methodik vor einiger Zeit bereits von der Arbeitsgruppe im Rahmen des Simulationssystems Dynamis IIX. Ziel der damaligen Eigenentwicklung war, den Aufwand der Modellerstellung und der Simulationsdurchführung für dieses Anwendungsgebiet zu verringern und die Modelle transparenter zu machen.

Dieser Beitrag beschreibt nun eine Neuimplementation der Methodik auf der Basis des offenen Simulationssystems EXTEND, wobei explizit aber der Erkenntnisgewinn aus der damaligen Entwicklung in die aktuelle Reimplementation eingeflossen ist.











Konstruktions-tabelle	Kombination von Tabellen durch die Mengenoperationen der Relationenalgebra	
Eingangstabelle	Laden externer Daten in das Modell; Prognose/Schätzung nicht verfügbarer Daten	
Schlüssel	Bestimmung eines Selektionsprädikates	
Selektor	Selektion von Zeilen einer Tabelle mit Hilfe eines Selektionsprädikates	
Ausgabe-Selektor	Darstellen einer ausgewählten Zeitreihe in Kurvenform	
Operationsvorschrift	Bildung eines arithmetischen Ausdrucks	
Operator	Arithmetische Verknüpfung von Spalten einer Tabelle	
Aggregator	Aggregation von Tabellenzeilen	
Submodel	Modulare (hierarchische) Modellierung	
Inout	Kommunikation zwischen Submodellen	

Tabelle 1
Modellkomponenten des Dynamic Relations-Ansatzes

2.1 Die Modellkomponenten von Dynamic Relations

Die Methodik Dynamic Relations soll den Aufbau von grafischen Modellen auf der Grundlage umfangreicher empirischer Datensammlungen ermöglichen. Um eine Verwendung dieser Daten im Rahmen eines Emissionsmodells zu unterstützen, enthält Dynamic Relations Komponenten, die die grafische Erstellung tabellenbasierter Modelle explizit unterstützen (vgl. Tab. 1).

2.1.1 Modellkomponente Datenselektion

Die Datenselektion ermöglicht es, bestimmte Zeilen einer Tabelle zu selektieren und nur diese Zeilen in die Simulation einfließen zu lassen. Die Selektion erfolgt über eine Selektionsbedingung, die angibt, welche Werte die Attribute eines Tupels besitzen dürfen. Durch Konjunktion und Disjunktion einzelner Selektionsbedingungen können komplexe Auswahlkriterien definiert werden.

2.1.2 Modellkomponente Datenprojektion

Die Datenprojektion ermöglicht die Auswahl bestimmter Spalten einer Tabelle. Die Projektionsbedingung wird durch die Angabe der in der Ergebnistabelle benötigten Spalten formuliert. Die dabei entstehende Redundanz der Tabellenzeilen muß anschließend bereinigt werden.

2.1.3 Modellkomponenten für Mengenoperationen

Die Eigenschaften der Systemelemente können über mehrere Tabellen verteilt sein. Es ist daher wichtig, die Tabellen kombinieren und damit in neue Beziehung setzen zu können. Da jede Tabelle eine Menge repräsentiert, können die Tabellen über die Mengenoperationen Vereinigung, Durchschnitt, Differenz und Kreuzprodukt verknüpft werden.

2.1.4 Modellkomponenten für arithmetische Verknüpfungen

Mit den bis jetzt beschriebenen Anfrageoperationen des Relationenmodells können lediglich Daten abgefragt werden. Es ist nicht möglich, aus ihnen neue Daten abzuleiten. Zu diesem Zwecke wird die arithmetische Verknüpfung von Tabellenwerten eingeführt. Die Verknüpfung wird durch eine Funktionsgleichung beschrieben und findet zeilenweise statt. Die Tabelle erhält eine weitere Spalte, deren zeilenbezogenen Werte durch Berechnung der Funktionsgleichung mit Werten der entsprechenden Zeile ermittelt wird (siehe auch Abb. 2).

2.1.5 Modellkomponenten zur Datenverdichtung

Durch Gruppenbildung und Datenverdichtung wird eine statistische Auswertung der Tabellendaten ermöglicht. Hierbei wird eine Gruppierungsspalte festgelegt. Zeilen einer Tabelle, welche in dieser Spalte die gleiche Ausprägung besitzen, werden gruppiert. Anschließend erfolgt mit Hilfe von Aggregatfunktionen eine Verdichtung dieser Zeilen zu einer Zeile. Die Anzahl der Zeilen der Ausgangstabelle ist also gleich der Anzahl der unterschiedlichen Ausprägungen in der Gruppierungsspalte. Als Aggregatfunktionen stehen zur Verfügung: Summe, Zählung, Minimum, Maximum, Durchschnitt, Streuung und Varianz.

2.1.6 Modellkomponenten zur Prognose

Zum Füllen von Datenlücken und zur Prognose sind in der Methode Dynamic Relations zwei Varianten vorgesehen. Es sind dies die *A-priori*-Prognose und die *Induktive* Prognose. Es ist möglich, die Methode um weitere Prognosen zu erweitern. Primär jedoch erschien es wichtig, die Machbarkeit dieser Prognosen zu zeigen und ihre Einbindung in die Methode darzustellen.

Die *A-priori*-Prognose: Bei dieser Prognose wird vorausgesetzt, daß der Modellentwickler erstens ein Wissen darüber hat, wie die Tabellenwerte sich in Abhängigkeit zur Simulationszeit ändern, und er zweitens die Fähigkeit besitzt, dieses Wissen in einer Funktionsgleichung darzustellen. Es wird also für jeden Tabellenwert eine zeitabhängige Funktionsgleichung definiert, anhand derer die Methode die Werte berechnen kann. Die Funktionsgleichungen werden in einer Funktionenmatrix angegeben, die für jeden Simulationsschritt ein Tabellensegment erstellt und dem Modell zur Verfügung stellt.

Die *Induktive* Prognose: Bei dieser Prognosekomponente wird davon ausgegangen, daß die Bedingungen der *A-priori*-Prognose nicht zutreffen. Die Methode berechnet die Tabellenwerte zum aktuellen Simulationszeitpunkt anhand der entsprechenden Tabellenwerte von vergangenen Simulationszeitpunkten. Die Anzahl der in die Berechnung einfließenden vergangenen Tabellenwerte ist variabel. Als ein Beispiel für die induktive Prognose ist in der Methode Dynamic Relations die lineare Regression implementiert. Sie berechnet eine Gerade, die nach dem Prinzip von Gauß den Punkten der Zeitreihe am besten angepaßt ist, wenn die Quadratsumme der Abstände zwischen dieser Geraden und den Punkten minimal ist.

3 Das Simulationswerkzeug EXTEND

Um nun zu überprüfen, inwieweit diese Modellierungsmethodik auf der Basis eines Standard-Simulationssystems implementiert werden kann, haben wir unter Verwendung von EXTEND Version 4 eine Bausteinbibliothek erzeugt, die die Verkehrsträgeremissionsmodellierung auf der Grundlage von Dynamic Relations unterstützt. Bei EXTEND handelt es sich um ein grafisches Modellierungswerkzeug für den Macin-

tosh als auch für PCs unter Windows, das käuflich am Markt erworben werden kann. EXTEND bietet sowohl eine grafische Modellerstellungskomponente als auch die Möglichkeit, mit Hilfe seiner C-ähnlichen Programmiersprache ModL (vgl. Abb. 1) eigene Erweiterungen in das Simulationswerkzeug zu integrieren.

Die Schlüsselidee von EXTEND ist aber sein bibliotheksbasiertes Blockkonzept. Aus mitgelieferten Standardbibliotheken entnimmt der Modellentwickler vorgefertigte Modellbausteine (sog. Blöcke) und baut hieraus sein Modell auf. Hierbei repräsentieren die Blöcke Aktionen bzw. Prozesse, welche während der Simulation ausgeführt werden. Die Blöcke besitzen Ein- und Ausgabeschnittstellen, Konnektoren genannt. Anhand dieser Konnektoren können die Blöcke miteinander verbunden und somit in Beziehung zueinander gesetzt werden. Es findet automatisch eine Kausalitätsprüfung dieser Verbindungen statt, wobei der Benutzer interaktiv auf Fehler aufmerksam gemacht wird. Über diese Verbindungen erfolgt somit der Datenaustausch zwischen den Blöcken.

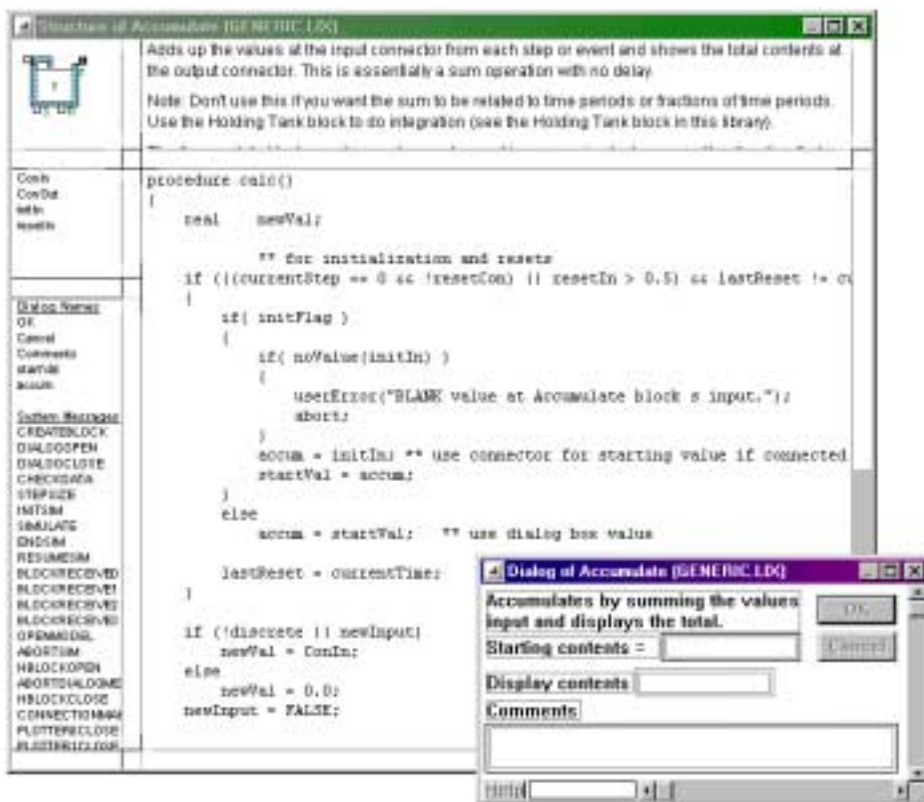


Abbildung 1
Programmskript und Dialogfenster eines Blocks in EXTEND

Der Datenaustausch erfolgt dabei üblicherweise vom Ausgangskonnector eines Blockes zu den Eingangsknectoren der mit ihm verbundenen Blöcke. Der Modellentwickler muß diesen Standarddatenaustausch nicht initiieren, da er vom System übernommen wird. Jeder Block verarbeitet die Daten an seinen Eingangsknectoren und stellt die Ergebnisse an seinen Ausgangsknectoren den nachgeschalteten Blöcken zur Verfügung. Die Verarbeitung erfolgt durch ein, zum Block gehöriges, Programmskript (siehe Abb. 1). In der Regel ist dieses Programmskript für den Modellentwickler nicht von Interesse, es sei denn, er möchte die nicht variablen Eigenschaften des Blocks gezielt verändern. Die kontextspezifischen Eigenschaften eines Blocks werden in einem dem Block zugeordneten Dialogfenster eingestellt, so daß es möglich ist, den Block im gewissen Rahmen an die Modellbedürfnisse anzupassen. EXTEND bietet dabei einen eigenen Dialogeditor an, mit dessen Hilfe der Anwender zu jedem Block eigene Dialoge entwickeln kann.

EXTEND unterstützt in beliebiger Tiefe eine hierarchische Erstellung von Modellen. Es ist möglich, ein Modell aus verschiedenen Teilmodellen zusammenzusetzen. Diese Teilmodelle bzw. Blöcke können in einer Bibliothek abgespeichert werden und jederzeit wieder aus dieser Bibliothek in einem anderen Modell eingesetzt werden. Auf diese Weise wird eine Wiederverwendung bereits erstellter Modellkomponenten oder gar ganzer Modelle optimal unterstützt. Dieser Hierarchisierungsmechanismus erhöht insbesondere in komplexen Modellen die Übersichtlichkeit merklich. Zusätzlich können eigene Blöcke erstellt oder Standardbibliotheken angeschlossen werden, die für bestimmte relevante Anwendungsbereich bzw. Simulationsmethoden allgemein benutzbare bzw. standardisierte Modellbausteine enthalten.

Sämtliche Blöcke von EXTEND werden in Bibliotheken verwaltet. Dabei enthält eine Bibliothek Blöcke mit ähnlichen Simulationsverhalten bzw. Aufgabenbereichen. Zu den wichtigsten Standardbibliotheken gehören die "Discrete Event"-, die "Generic"- und die "Plotter"-Bibliothek, wobei die erste spezielle Blöcke zur Simulation diskreter Ereignisse, die zweite zur Simulation kontinuierlicher Modelle und die letzte verschiedene Blöcke zur Ausgabe und Analyse der Simulationsergebnisse enthält. Die Modelle können aus Blöcken unterschiedlicher Bibliotheken erstellt werden. Somit handelt es sich bei EXTEND um ein recht offenes Simulationstool, mit dem kontinuierliche, diskrete oder kombinierte Simulationsmodelle erstellt werden können.

Die Durchführung von Simulationsexperimenten wird von EXTEND ebenfalls auf vielfältige Weise unterstützt. Neben elementaren Funktionen steht zum Beispiel eine Sensitivitätsfunktion zur Verfügung. Mit ihr ist es möglich, mehrere Simulationen hintereinander durchzuführen, bei denen die Parameter systematisch variiert werden können. Zur Laufzeit einer Simulation erhält der Benutzer Informationen durch Report- und Trace-Funktionen. Die Simulation kann jederzeit unterbrochen und mit geänderten Parametern fortgesetzt werden. Eine weitere Möglichkeit ist die Animationsfähigkeit der Blöcke, welche ihr Erscheinungsbild in Abhängigkeit von internen Zustandsvariablen verändern können. Dieses ermöglicht es dem Benutzer, die Si-

mulationsergebnisse im Groben direkt im Modell zu analysieren, ohne expliziert die Report-Funktion anzuschalten. Für weitere Detailinformationen über EXTEND wird auf (Imagine That! 1997) oder (Krahl 1995) verwiesen.

4 Implementation von Dynamic Relations mit EXTEND

Die Implementation von Dynamic Relations mit EXTEND ist durch Erstellung einer neuen Bibliothek erfolgt, in der die verschiedenen Funktionen der Methodik durch jeweils einen Block realisiert sind (Klinger 1999).

Die Funktionalität der erstellten Blöcke wird dabei in der EXTEND-eigenen Programmiersprache ModL definiert. Hierbei wird intensiv davon Gebrauch gemacht, daß das Simulationssystem zu bestimmten Simulationszeitpunkten gezielt spezielle Nachrichten an die einzelnen Blöcke sendet, worauf diese entsprechende Prozeduren ausführen. So initialisieren die Blöcke der Dynamic Relations-Bibliothek beim Erhalt der Nachricht "CreateBlock" wichtige Anfangsvariablen. Außerdem werden die Default-Werte der Dialog-Fenster eingestellt. In Abb. 2 ist beispielhaft das Dialogfenster des Operatorblocks abgebildet.

Damit der Benutzer schon vor der ersten Simulation im Dialogfenster die variablen Eigenschaften des Blocks an das Modell anpassen kann, müssen dem Block gewisse Tabellendaten zur Verfügung stehen. Aus diesem Grund werden zusätzlich noch Default-Werte dieser Tabellendaten generiert, welche nach dem ersten Simulationsschritt durch "richtige" Daten der Eingangstabelle ersetzt werden.

Nach dem Start einer Simulation, aber noch vor dem ersten Simulationsschritt erhalten die Blöcke des Modells vom Simulationssystem die Nachrichten "CheckData" und "InitSim". Da die Dynamic Relations-Blöcke keine Prozedur "on CheckData" erhalten, wird die "CheckData"-Nachricht ignoriert. Die "InitSim"-Nachricht hingegen bewirkt die Reservierung des Speicherplatzes zur Aufnahme der Tabellenstruktur sämtlicher benötigter Tabellen des Blocks. Hiernach ist der Block bereit für die weiteren Simulationsschritte.

Zum Ausführen eines Simulationsschrittes sendet das Simulationssystem eine "Simulate"-Nachricht an die Blöcke des Modells. Jeder Block reagiert darauf durch Einlesen der Pointer an seinen Eingangskonnektoren. Die Größe der Tabellen, auf die die Pointer referenzieren, wird ermittelt. Der Größe entsprechend reserviert der Block den Speicherplatz zur Aufnahme der kompletten Tabellenstrukturen, um eine Kopie der Eingangstabellen anzulegen. Die Verarbeitung der Tabellen erfolgt auf diesen Kopien, so daß Änderungen an diesen Tabellen nicht die Eingangstabelle verändern. Dieses ist insofern von Bedeutung, da eine Tabelle die Eingangstabelle mehrerer Blöcke sein kann und jeder Block zur Verarbeitung die ursprüngliche Tabelle benötigt. Nachdem die Arbeitskopien der Eingangstabellen angelegt sind, erfolgt die eigentliche Verarbeitung der Tabellen anhand der entsprechenden Verarbeitungsvorschriften, welche sich aus der Blockdefinition von Dynamic Relations ergeben. Nach

erfolgter Verarbeitung wird noch ein Pointer auf die Ergebnistabelle (Ausgangstabelle) an den Ausgabekonnektor angelegt, um den nachfolgenden Blöcken die Weiterverarbeitung der Tabelle zu ermöglichen.

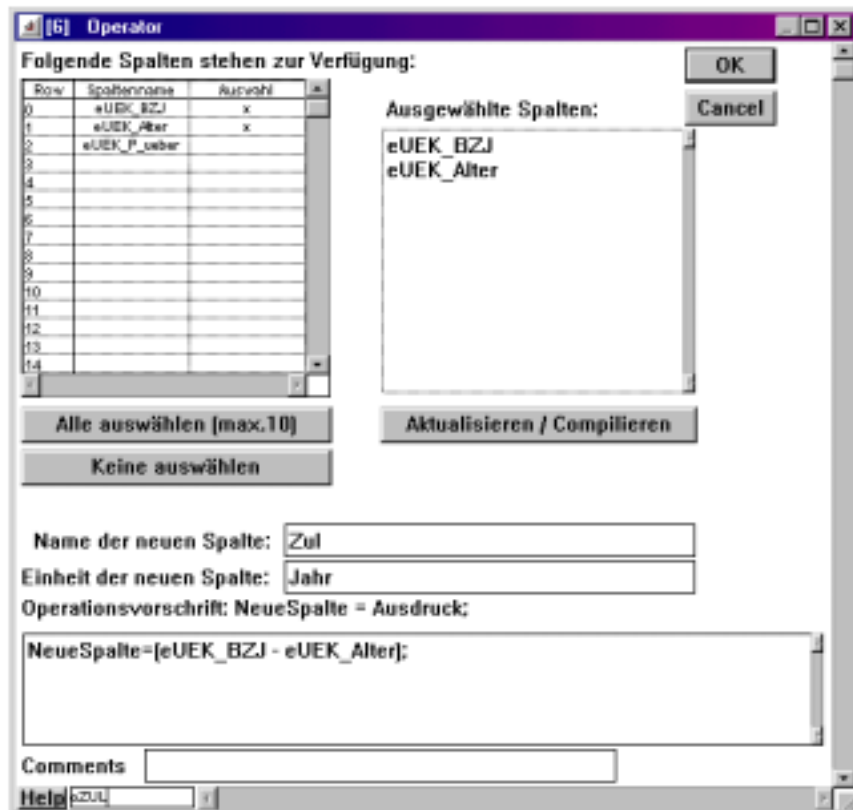


Abbildung 2
Dialogfenster des Operatorblocks

Die Simulation endet mit einer "EndSim"-Nachricht, welche die Blöcke veranlaßt, den für die Simulation reservierten Speicherplatz wieder frei zu geben.

Die erstellten EXTEND-Bausteine besitzen somit eine gewisse Kommunikationsfähigkeit, indem sowohl während der Simulation als auch im Vorfeld eines Simulationslaufes bestimmte Nachrichtentypen untereinander ausgetauscht werden können. Für diese Nachrichtentypen können dann vom Programmierer einzelne Simulationsroutinen bereitgestellt werden, so daß eine blockindividuelle Reaktion auf diese Nachrichten möglich ist. Dieser Mechanismus ist insbesondere von uns dazu verwendet worden, um den Benutzer während der Modellbildungsphase interaktiv auf

Inkonsistenzen im Modell hinzuweisen. Der Anwender kann deshalb gezielt Blöcke öffnen, für die ihm vom System Inkonsistenzen gemeldet worden sind. Im geöffnetem Fenster findet er die aktuellen Tabellenspalten vor, da sich der Block während des Öffnens aktualisiert. Diese automatische Aktualisierung greift auf EXTEND-Routinen zur Blockkommunikation zurück. Hierbei hat jeder Block die Möglichkeit, eine Nachricht an die ihm nach- oder vorgeschalteten Blöcke zu senden, so daß die die Nachricht empfangenden Blöcke darauf reagieren können.

In Abb. 3 ist im linken Fenster die Dynamic Relations-Bibliothek unter EXTEND zu sehen. In ihr enthalten sind die Bausteine der Methodik, aus denen das rechts daneben stehende Beispielsmodell erstellt wurde. Eine mögliche Ergebnispräsentation ist in Form eines Plots im unteren Fenster zu sehen.

Ferner zeigt die Abbildung ein exemplarisches Modell zur Berechnung der Schadstoffemissionen eines bestimmten KFZ-Typs, welches mit Hilfe unserer implementierten Bibliothek aufgebaut wurde. Die Daten zur Ermittlung der Emissionen sind auf verschiedene Tabellen verteilt. Durch den Verbund-Block "Verb_Zul" werden die drei Tabellen "NZZ" (Neuzulassungszahlen), "NZA" (Neuzulassungsanteil) und "UEK" (Überlebenskurven) zusammengefügt, nachdem zwei Tabellen durch Selektion ("Ausw_KFZ_Typ") und Operation ("oZUL") angepaßt wurden. Die durch den Verbund resultierende Tabelle enthält u.a. Informationen über die Anzahl aller zugelassenen KFZ-Typen pro Zulassungsjahr, den prozentualen Anteil des zu untersuchenden KFZ-Typs sowie die Wahrscheinlichkeit dieses KFZ-Typs, noch im Straßenverkehr benutzt zu werden (sog. „Überlebenswahrscheinlichkeiten“). Aus diesen Informationen wird durch Multiplikation im Operator-Block "opBest" die Information "Bestand nach Alter" des KFZ-Typs gewonnen. Die hieraus resultierende Tabelle wird im Verbund-Block "Verb_Fahrstg" mit den Tabellen "MFL" (Mittlere Fahrleistung) und "STR" (Strassenartenanteile) zusammengeführt, wodurch Informationen über die Jahresfahrleistung in Abhängigkeit zum KFZ-Alter ermittelt werden können. Die nun folgenden Funktionen sind in der Abb. 3 nicht dargestellt (sondern werden dort durch usw. gekennzeichnet), nämlich eine Operation zur Ermittlung der gesamten Jahresfahrleistung unabhängig vom Alter, ein Verbund mit einer Tabelle mit Informationen der Emissionsfaktoren in Abhängigkeit von Straßenart, Alter, Typ und Minimierungstechnologien, sowie drei weitere Operationsblöcke zur Berechnung der Emissionen NO_x, HC, und CO pro Alterstyp.

Die resultierende Tabelle wird im Aggregations-Block "Aggregation" mit der Spalte "Bezugsjahr" als Gruppierungsattribut zusammengesetzt. Es folgt eine Projektion auf relevante Spalten, welche über den Schnittstellen-Block DR2Plotter auf den Plotter-Block ausgegeben werden. Parallel hierzu erfolgt eine Speicherung der Ergebnisse durch den Ausgangstabellen-Block "Ergebnis".

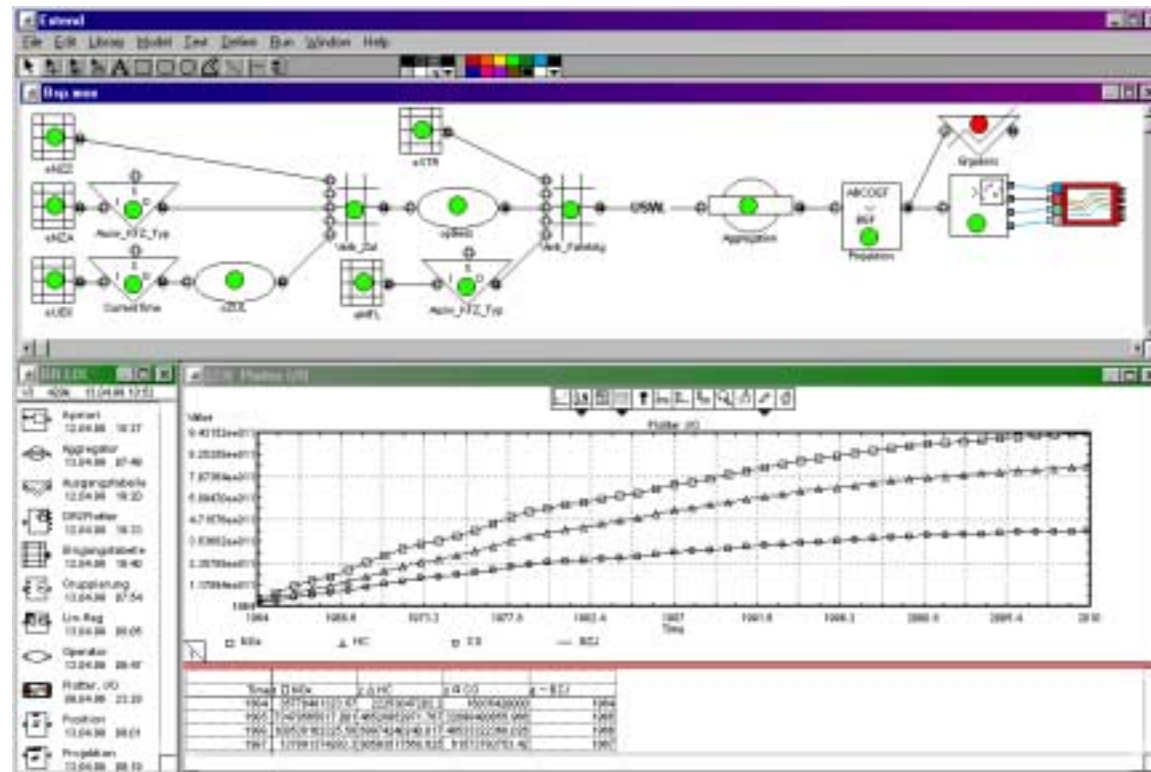


Abbildung 3
Dynamic Relations in EXTEND

5 **Schlußfolgerungen und Ausblick**

Die Implementation von Dynamic Relations mit Hilfe von EXTEND erwies sich als gut. Es wurden einige wenige Funktionen vermißt, welche jedoch die Funktionalität der Methodik nicht beschränken. Vielmehr könnte sogar die Mächtigkeit der Methodik mit Hilfe von EXTEND aufgrund seiner Offenheit erweitert werden. So ist z.B. die interaktive Inkonsistenzprüfung erst durch die interne Blockkommunikationsfähigkeit realisierbar. Außerdem konnte der Funktionsumfang einiger Modellbausteine im Vergleich zur originären Dynamic Relations-Methodik durch Einbindung der ModL-Syntax erweitert werden. Neu ist außerdem die Visualisierung der Gültigkeit eines Blocks durch die Animationsfunktionalität von EXTEND.

Bezugnehmend auf die eingangs gestellte Frage, läßt sich auch für das in diesem Beitrag beschriebene Anwendungsgebiet zusammenfassend festhalten, daß die Verwendung eines offenen Standardsimulationssystems wie EXTEND als Basis für eine Methodenentwicklung gut geeignet erscheint. So ist die Entwicklung einer die Dynamic Relations-Methodik umsetzenden Bibliothek in EXTEND erheblich schneller und mit weniger Aufwand erfolgt als unsere damalige Eigenentwicklung. Außerdem konnte sogar die originäre Methodik in seinem Funktionsumfang durch das neu entwickelte Bibliothekensystem in einigen Punkten recht einfach erweitert werden. Wir erwarten des weiteren einen geringeren Wartungsaufwand und einen geringeren Einarbeitungsaufwand bei der Entwicklung zukünftiger Erweiterungen. Insgesamt erscheint uns das neue System auch stabiler in seinem Laufzeitverhalten zu sein als die damalige Applikation. Betont sei in diesem Zusammenhang, daß die erwähnten Vorteile nur aufgrund der Offenheit und leichten Erweiterbarkeit von EXTEND zum Tragen kommen. Bei einer Verwendung eines weniger offenen Standard-Simulationssystems als Basis einer Entwicklung wäre die Aussage für die Verwendung von Standardsoftware anstelle einer Individualentwicklung nicht so klar ausgefallen.

Ein weiterer Ansatzpunkt unserer Arbeiten wird die Übertragung dieser grafischen Modellierungsmethode auf ihre allgemeine Anwendbarkeit hin darstellen. Es ist denkbar, daß die vorgestellte Methodik für andere Anwendungsgebiete, die auf der Basis tabellenorientierter Zeitreihen arbeiten, benutzt werden kann. Dieses werden wir in nächster Zeit verifizieren. Insofern sehen wir hier eine Chance, daß eine speziell für die Umweltinformatik entwickelte Methodik in anderen Bereichen der Angewandten Informatik Verwendung findet.

Literaturverzeichnis

- Freese, H., Häuslein, A., Isbarn, I., Klee, A., Niederle, W., Page, B., Seidel, J. (1994): Ein Werkzeug zur Modellierung der Verkehrsträgeremissionen auf der Basis einer tabellenorientierten, grafischen Simulationstechnik, in: Hilty, L.M., Jaeschke, A., Page, A., Schwabel, A. (Hrsg.): Informatik für den Umweltschutz, 8. Symposium Hamburg, Marburg, S. 327 - 334
- Freese, H., Seidel, J. (1999): Entwicklung einer grafischen, tabellenbasierten Simulationsmethodik zur Emissionsmodellierung, Diplomarbeit, Fachbereich Informatik, Universität Hamburg
- Grünewald, A. (1997): Entwicklung einer EXTEND-Modellbausteinbibliothek für die Simulationemethode System Dynamics, Studienarbeit, Fachbereich Informatik, Universität Hamburg
- Klinger, D. (1999): Realisierung der grafischen Modellierungsmethode "Dynamic Relations" für zeitreihenbasierte Modelle auf der Basis des offenen Simulationssystems EXTEND, Studienarbeit, Fachbereich Informatik, Universität Hamburg
- Krahl, D. (1995): Building End User Applications with EXTEND, in: Alexopoulos et al. (Hrsg.): Proceedings 1995 Winter Simulation Conference, S. 413-419
- Imagine That! Inc. (1997): EXTEND - Simulation Software for the next Millennium, User's Manual for EXTEND, Version 4
- Niederle, W. (1991): „Rapid Prototyping“ von Verkehrsemissionsmodellen, in: Hälker, M., Jaeschke, A. (Hrsg.): Informatik für den Umweltschutz, Berlin et al., S. 11 - 20
- Page, B., Kreutzer, W., Wohlgemuth, V., Brüggemann, R. (1997): Ein Anwendungsvergleich ausgewählter graphischer Modellierungswerkzeuge in der Expositionsanalyse von Chemikalien in der Umwelt, in: Grützner, R. (Hrsg.): Modellierung und Simulation im Umweltbereich: Fortschritte in der Simulationstechnik, Braunschweig/Wiesbaden, S. 147 - 172
- Theilmann, C., Wolter M. (1999): Erweiterung eines Bibliothekensystems zur Expositionsanalyse von Chemikalien in der Umwelt, Studienarbeit, Fachbereich Informatik, Universität Hamburg
- Wohlgemuth, V. (1997): Simulationswerkzeuge zur Modellierung des Transports und Verbleibs von Chemikalien in der Umwelt: Anforderungsanalyse, Bestandsaufnahme, Modellimplementation und konzeptionelle Erweiterung, Diplomarbeit, Fachbereich Informatik, Universität Hamburg