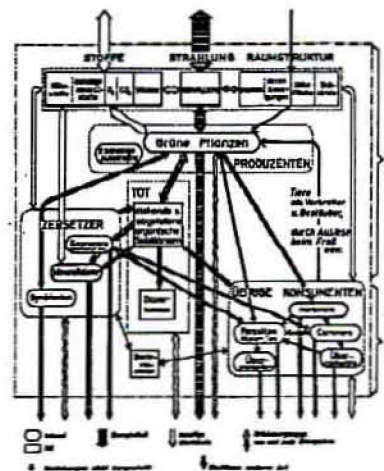


Fallstudie interdisziplinäre Modellierung eines terrestrischen Ökosystems unter Einfluss des Treibhauseffektes

Andreas Fischlin und Olivier Roth

sowie

T. Blanke, H. Bugmann, D. Gyalistras und F. Thommen



März / March 1990

Eidgenössische Technische Hochschule Zürich ETHZ
Swiss Federal Institute of Technology Zurich

Departement für Umweltnaturwissenschaften / Department of Environmental Sciences
Institut für Terrestrische Ökologie / Institute of Terrestrial Ecology

The System Ecology Reports consist of preprints and technical reports. Preprints are articles, which have been submitted to scientific journals and are hereby made available to interested readers before actual publication. The technical reports allow for an exhaustive documentation of important research and development results.

Die Berichte der Systemökologie sind entweder Vorabdrucke oder technische Berichte. Die Vorabdrucke sind Artikel, welche bei einer wissenschaftlichen Zeitschrift zur Publikation eingereicht worden sind; zu einem möglichst frühen Zeitpunkt sollen damit diese Arbeiten interessierten LeserInnen besser zugänglich gemacht werden. Die technischen Berichte dokumentieren erschöpfend Forschungs- und Entwicklungsergebnisse von allgemeinem Interesse.

Adresse der Autoren / Address of the authors:

Systemökologie ETHZ
Institut für Terrestrische Ökologie
Grabenstrasse 3
CH-8952 Schlieren/Zürich
SWITZERLAND
e-mail: sysecol@ito.umnw.ethz.ch

Vorbemerkung

Im Verlaufe des Sommersemesters 1989 hat die Fachgruppe Systemökologie zusammen mit den Studierenden der Abteilung für Umweltnaturwissenschaften an der Abteilung für Umweltnaturwissenschaften XB der ETH Zürich die Fallstudie, die in diesem Bericht beschrieben wird, bearbeitet. Der Versuch war ein Experiment, bei dem sowohl die neu gebildete Fachgruppe wie die Studierenden das erste Mal mit der Problematik einer gemeinsamen Modellierung konfrontiert worden sind. Die Studierenden besuchten die für sie obligatorische zweistündige Vorlesung Systemanalyse II (Naturwissenschaftliche Anwendungen) im Anschluss an die Veranstaltung Systemanalyse II (Allgemeine Systemtheorie), die sie im vorhergehenden Wintersemester besucht hatten.

Modellierungen im Team sind ein seit langem viel geforderter aber häufig schwer zu befriedigender Traum interdisziplinärer Forschungsprojekte (LEVINS, 1966; HOLLING, 1966; WATT, 1968; NORTON & HOLLING, 1979). Besondere Schwierigkeiten entstehen oft durch die Tatsache, dass Experten dazu neigen, ihre jeweiligen Spezialitäten in den Vordergrund zu stellen, ungeachtet der Tatsache, ob diese Vorliebe von den anderen Spezialisten geteilt werden kann oder nicht (ZIMAN, 1987) und oft ohne zu bemerken, dass die für den Spezialisten zurzeit vordringlichsten und interessantesten Fragen unter Umständen zur Lösung des gemeinsamen Zieles kaum etwas bis gar nichts beizutragen vermögen. Häufig erfordert eine bestimmte Fragestellung, ganz klare Randbedingungen, die erfüllt sein müssen, damit ein Team von Spezialisten überhaupt koordiniert miteinander forschen kann. Durch diese Situation haben schon häufig mit viel Schwung und Enthusiasmus in Angriff genommene Projekte entweder gedroht im Sande zu verlaufen, oder sie haben ihr ursprüngliches Ziel wenn überhaupt nur noch unter äussersten Anstrengungen erreichen können (HILBORN, 1981; CUFF & TOMCZAK, 1983; ELLENBERG, 1986).

Die Strategie die in dieser Arbeit eingesetzt worden ist, versucht nun derartige Konflikte nicht im Nachhinein zu lösen, sondern sie erst gar nicht aufkommen zu lassen. Sie beruht auf der Technik der Ausschau-Matrix (outward looking matrix) und kommt der Psychologie und Arbeitsweise von Experten sehr entgegen. Viele Studien dieser Art beispielsweise in Nordamerika (ESSA, 1982) haben gezeigt, dass mit dieser Vorgehensweise, insbesondere im Umweltbereich etwa bei Umweltverträglichkeitsprüfungen, sehr gute Erfahrungen gemacht werden können. Es ist bei entsprechend sorgfältiger Planung möglich, gezielt auf das gemeinsame Projektziel hin zu arbeiten und Modelle zu entwickeln (ESSA, 1982; WALTERS, 1986).

Bei dieser Vorgehensweise erfüllen die dabei entstehenden Modelle mehrere Rollen: Sie fördern den Erfahrungs- und Wissensaustausch zwischen den beteiligten, meist disziplinenorientiert zusammengesetzten Arbeitsgruppen und können vielfach gleichzeitig eines der wichtigen, gemeinsamen Arbeitsziele darstellen. Für Wissenschaftler ist das häufig eine besonders stimulierende Herausforderung an das eigene Können und Wissen. Allerdings soll hier auch nicht der Eindruck erweckt werden, dass die Arbeitstechnik der Ausschau-Matrix das interdisziplinäre Arbeiten nun aller Schwierigkeiten entledigen würde. Jedoch ermöglichen diese Techniken, dass die häufigsten Schwierigkeiten bei geschicktem Anpacken, insbesondere auch «weiser» Voraussicht (HOLLING, 1981; CLARK & MUNN, 1986) durch erfahrene Projektleiter vermieden werden können. Im weiteren zeigt die Erfahrung ebenfalls, dass Modellierung in der vorgeschlagenen Art ein wichtiges Hilfsmittel zur Förderung disziplinübergreifenden Arbeitens sein kann, zumindest so wie der Sauerteig auch mithilfe aus Teig schmackhaftes Brot zu machen.

In diesem Sinne bitten wir Sie, den beiliegenden Text zu verstehen. Nicht das resultierende Modell stand im Vordergrund, sondern der Weg seiner Herleitung. Wir erheben deshalb keinen Anspruch auf die Korrektheit des Modelles; auch soll den erzeugten Simulationsresultaten kein ihnen nicht zustehendes Gewicht zugemessen werden. Es ging uns darum pilotprojektartig aufzuzeigen und zu erproben – mit einer

derart grossen Zahl von Beteiligten (über 100) ist bis jetzt so etwas u.E. noch nie durchgeführt worden –, wie in etwa ein Modell zur Beantwortung der aufgeworfenen Frage nach der Reaktion ein ganzes Ökosystem unter veränderten Klimabedingen multidisziplinär entwickelt werden könnte. Es wurde an einem konkreten Fallbeispiel, den Auswirkungen des Treibhauseffektes auf ein schweizerisches Agroökosystem im Jahre 2050, gezeigt, wie und mit welchen Techniken eine grosse Zahl von Leuten durch ihren Gruppenbeitrag zum Gelingen eines gemeinsamen Projektes fruchtbar beitragen können. Das Beispiel erforderte Beiträge aus den Bereichen Entwicklung der wirtschaftlichen Tätigkeiten im globalen Massstab in bezug auf CO₂-Emmissionen bezüglich dem Treibhausklima, des Jahresverlaufes der Witterung unter veränderten klimatischen Bedingungen und den dadurch verursachten Reaktionsänderungen an einem ganzen Ökosystem mit Vertretern aus seinen wichtigsten funktionellen Gruppen wie den Primärproduzenten (Pflanzen), Herbivore und Karnivore (Tieren) und den Destruenten (Bodenorganismen insbesondere Bakterien).

Ich möchte an dieser Stelle jedoch die Gelegenheit benutzen, um allen an diesem Projekt massgeblich beteiligten Mitarbeitern, Assistenten und vor allem auch den Studierenden zu danken.

Zürich, den 9. März 1990

A. Fischlin

Cited Literature

- BENYON, P., 1972. *Computer modelling and interdisciplinary teams*. Search 3 (7): 250-6.
- CLARK, W.C. & MUNN, R.E. (EDS.), 1987. *Sustainable development of the biosphere*. Cambridge University Press, Cambridge a.o., 491pp.
- CUFF, W. & TOMCZAK, M. (eds.), 1983. *Synthesis and modelling of intermittent estuaries - A case study from planning to evaluation*. Lecture Notes on coastal and estuarine studies, Vol. 3. Springer, Berlin a.o.,
- ELLENBERG, H., MAYER, R. & SCHAUERMANN, J. (EDS.), 1986. *Ökosystemforschung. Ergebnisse des Sollingprojekts 1966-1986*. Eugen Ulmer, 507pp.
- ESSA ENVIRONMENTAL AND SOCIAL SYSTEMS ANALYSTS LTD., 1982. *Review and Evaluation of Adaptive Environmental Assessment and Management*. Environment Canada, Minister of Supply and Services Canada 83 Cat. No. En21 - 36/1983E, 116pp.
- HILBORN, R., 1981. *Some failures and successes in applying systems analysis to ecological systems*. J. Appl. Systems. Analysis, 6: 25-31.
- NORTON, G.A. & HOLLING, C.S. (EDS.), 1979. *Pest management*. Proc. of an Internat. Conference, October 25-29, 1976, (Vol. 4, IIASA (International Institute for Applied Systems Analysis) Proceedings Series), Pergamon Press, Oxford, 352 pp.
- HOLLING, C.S., 1966. *The strategy of building models of complex ecological systems*. In: *Systems analysis in ecology*, K.E.F. Watt (ed.), Academic Press, New York.

- HOLLING, C.S., 1981. *Science for public policy: Highlights of adaptive environmental assessment and management*. Report R-23, Institute of Resource Ecology, UBC, Vancouver, B.C., Canada.
- LEVINS, R., 1966. *The strategy of model building in population biology*. Amer. Sci., 54: 421-31.
- WALTERS, C., 1986. *Adaptive management of renewable resources*. New York ; London : Macmillan, 374pp.
- WATT, K.E.F., 1968. *Ecology and resource management, a quantitative approach*. McGraw-Hill Book Company, New York, San Francisco, St. Louis, Toronto, London, sidney, 450pp.
- ZIMAN, J., 1987. *Knowing everything about nothing. Specialization and change in scientific careers*. Cambridge University Press: Cambridge a.o., 196pp.

Inhalt

VORBEMERKUNG

1	EINLEITUNG.....	1
2	SKIZZIERUNG DER FALLSTUDIE.....	2
3	KURZPORTRÄTS DER UNTERSISTEME.....	3
3.1	PRIMÄRPRODUZENTEN.....	3
3.2	HERBIVORE.....	3
3.3	KARNIVORE.....	3
3.4	DESTRUENTEN.....	4
3.5	WITTERUNG.....	4
3.6	TREIBHAUSKLIMA.....	4
4	ZUM VORGEHEN UND ARBEITSPLAN.....	4
5	ÜBERSICHT GESAMTMODELL.....	6
6	AUSSCHAUMATRIX.....	8
8	UNTERMODELLE.....	12
8.1	PRIMÄRPRODUZENTEN.....	12
8.2	HERBIVORE.....	14
8.3	KARNIVORE.....	16
8.4	DESTRUENTEN.....	16
8.5	WITTERUNG.....	18
8.6	TREIBHAUSKLIMA.....	19
9	RESULTATE.....	20
10	LITERATUR.....	21
A	GESAMTMODELL - PROGRAMMODUL (HAUPTPROGRAMM) TERRECOSSYS.....	I
B	DEFINITIONSMODULE.....	II
	PRIMÄRPRODUZENTEN - MODUL PRIMPROD.....	II
	HERBIVORE - MODUL HERBIVORES.....	II
	KARNIVORE - MODUL CARNIVORES.....	III
	DESTRUENTEN - MODUL DESTRUENTS.....	III
	WITTERUNG - MODUL WEATHER.....	III
	TREIBHAUSKLIMA - MODUL GREENHCLIM.....	IV
C	IMPLEMENTATIONSMODULE.....	IV
	PRIMÄRPRODUZENTEN - MODUL PRIMPROD.....	IV
	HERBIVORE - MODUL HERBIVORES.....	VI
	KARNIVORE - MODUL CARNIVORES.....	VIII
	DESTRUENTEN - MODUL DESTRUENTS.....	IX
	WITTERUNG - MODUL WEATHER.....	X
	TREIBHAUSKLIMA - MODUL GREENHCLIM.....	XIII

Fallstudie Modellierung eines terrestrischen Ökosystems unter Einfluss des Treibhauseffektes

1 Einleitung

In Form eines kleinen Miniprojektes soll die Modellierung eines ganzen terrestrischen Ökosystemes vorgenommen werden. Dieses Ökosystem soll insbesondere im Hinblick auf allenfalls stattfindende Veränderungen, z.B. in der Produktivität, aufgrund veränderter Klimabedingungen, wie sie durch den Treibhauseffekt prognostiziert werden (Schneider, 1989), untersucht werden. Allerdings soll weder die fachliche Auseinandersetzung mit der Biologie oder der Biometeorologie noch die detaillierte Auseinandersetzung mit den komplexen mathematischen Teilmodellen überhand nehmen. Im Vordergrund steht die Vorgehensweise, die exemplarisch am Beispiel zweier relativ einfacher Agroökosysteme eingeübt werden soll. Die folgenden Ziele werden verfolgt:

- Kennenlernen der Modellierung biologischer Prozesse
- Organisationsformen bei der Modellierung im Team von Spezialisten
- Simulationstechnik

Kennenlernen der Modellierung biologischer Prozesse: Biologische Prozesse sind gekennzeichnet durch eine grosse Vielfalt und relativ grosse Variabilität. Zufallsereignisse spielen überall deutlich erkennbar mit und können darunterliegende Mechanismen und Strukturen trotz ihrer Anwesenheit manchmal bis zur Unkenntlichkeit verwischen. Diese Tatsachen stellen an die Modellierung derartiger Prozesse besondere Anforderungen. Hierunter gehören die Fähigkeit Grenzen und Annahmen die in mathematischen Gleichungen stecken, genau zu erfassen und im Hinblick auf die vorhandenen biologischen Kenntnisse gut beurteilen zu können. Ökosysteme sind immer sehr komplexe Gebilde, die nie isomorph auf ein mathematisches Modell abgebildet werden können. Vielmehr ist es erforderlich ein im Hinblick auf die Fragestellung vereinfachtes Modell zu bilden. Letzteres ist zwar notwendig für den Erfolg, bringt jedoch auch den Nachteil mit sich, dass für neue Fragestellungen Modelle immer wieder neu gebildet werden müssen. Das erfordert nicht nur gründliches Sachwissen über die zu modellierenden Systeme, sondern auch die Fähigkeit Modellgleichungen aus bekannten Sachverhalten selbst herleiten zu können. Letzteres steht in diesem Teil der Vorlesung im Vordergrund.

Organisationsformen bei der Modellierung im Team von Spezialisten: Das Arbeiten im Team, sei sie inter- oder multidisziplinär, stellt immer besondere Anforderungen an die Teilnehmer. Die Systemanalyse hat in der Vergangenheit schon oft bei der Koordinierung und der Synthese von Spezialwissen eine besondere Rolle gespielt. Allerdings können dadurch auch Probleme und besondere Schwierigkeiten auftreten, die insbesondere im Umweltbereich manchmal zu derart entscheidender Bedeutung anwachsen, dass sie für Erfolg oder Misserfolg ausschlaggebend sind. Es soll eine in der Praxis schon seit vielen Jahren, insbesondere im Bereich der interdisziplinären Umweltforschung bewährte Technik, die Ausschau matrix oder «Looking Outward Matrix» kennengelernt und eingeübt werden. Dabei ist es wichtig, dass Sie selbständig in Gruppen arbeiten und dass die Gruppenarbeiten alle miteinander koordiniert und abgestimmt sind, so dass gegen das Ende eine Synthese der in den Gruppen entwickelten Teilmodlle zu einem Gesamtmodell wirklich vorgenommen werden kann.

Simulationstechnik: Komplexe, nichtlineare Modelle sind analytisch mit vertretbarem Aufwand kaum mehr bearbeitbar. Um trotzdem nicht auf bloss simplizistische Modelle

und Modellgleichungen beschränkt zu bleiben, werden komplexere, nichtlineare Modelle mit Hilfe von Rechnern numerisch untersucht. Eine zentrale Stellung nimmt hierbei die Lösung von Differentialgleichungs- und oder Differenzgleichungssystemen ein. Es soll eine auf modernen Arbeitsplatzrechnern, interaktiv einsetzbare Simulationstechnik kennengelernt werden, die für die Modellierung im Team besonders gut geeignet ist. Die hierzu notwendige Software, ModelWorks, wird zur Verfügung gestellt.

2 Skizzierung der Fallstudie

In Mitteleuropa und insbesondere in der Schweiz, besteht der grösste Teil der Landschaft, aus terrestrischen Ökosystemen. Für die Schweiz gilt: 28.3% Landwirtschaftliche Nutzfläche (Wies- und Ackerland, Obst- und Rebbau); 20.6% Weideland; 25.5% Wald; 21.3% Öd- und Unland inklusive Gebirge, Seen und Flüsse; 4.3% Überbautes Gebiet (Siedlungen, Industrie- und Verkehrsanlagen) (Bundesamt für Statistik, Stand 1972, Statistisches Jahrbuch der Schweiz 1989). Da es im Rahmen dieser Minifallstudie unmöglich ist die gesamte Biosphäre der Schweiz sinnvoll zu modellieren, wird ein konkreter Teil, d.h. ein einzelnes Ökosystem ausgewählt, um exemplarisch die Modellierung vorzunehmen (Fig. 1).

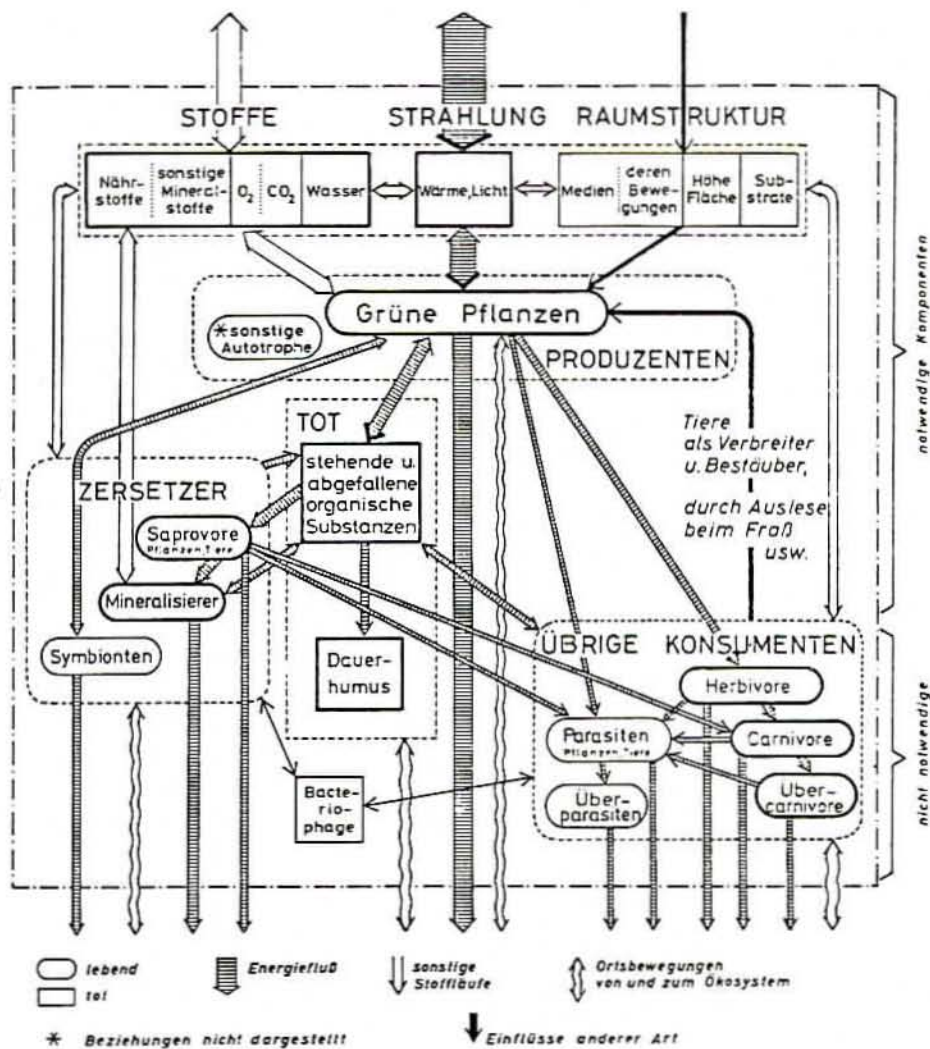


Fig. 1: Modell eines vollständigen Ökosystems nach Ellenberg, 1973

Entscheidend ist hierbei dass jedeR StudentIn den Prozess der Vereinfachung vom komplexen Vorbild zum homomorphen Modell selbst vollzieht. Die landwirtschaftlich genutzte Fläche dominiert, und weltweit gehören Weizen und Mais zu den wichtigsten Feldfrüchten der Welt (Weltweit Weizen 15.6%, Schweiz 31% (Total Winter u. Sommerweizen); Weltweit Mais 14.3%, Schweiz 21% (Körner- und Silomais)). Die Fallstudie soll sich deshalb mit den Agroökosystemen Mais und Weizen in der Schweiz heute und im Vergleich mit dem Jahr 2050 unter veränderten Klimabedingungen, insbesondere erhöhter Temperatur, auseinandersetzen. Als Anregung und Gedankenstütze bei der Modellierung des Ökosystems diene die Fig. 1.

3 Kurzporträts der Untersysteme

3.1 PRIMÄRPRODUZENTEN

Agroökosysteme sind in der Schweiz meist relativ einfache Ökosysteme, die auf der trophischen Stufe der Primärproduzenten relativ leicht zu erfassen sind. Die Biomasse des Agroökosystems Mais besteht hauptsächlich aus der Art *Zea Mays* L., das des Weizens aus der Art *Triticum aestivum* L. Die Nettoprimärproduktion gemessen als Zuwachs an pflanzlicher Biomasse, z.B. als Trockengewicht, in einem Agroökosystem wird immer durch abiotische Faktoren, insbesondere Temperatur und Strahlung, beeinflusst. Falls keine anderen Faktoren wesentlich eingreifen, spricht man auch vom Produktionsniveau 1 (Penning deVries & van Laar, 1982). Die Modellierung der Photosynthese und des Pflanzenwachstums innerhalb einer Vegetationsperiode kann unter diesen Bedingungen mit Hilfe bekannter Modelle, z.B. SUCROS (Penning deVries & van Laar, 1982), vorgenommen werden. Pflanzen sind sogenannte offene Organismen, d.h. es ist schwierig ein Individuum überhaupt abzugrenzen. Deshalb wird die Vegetationsdecke häufig nicht in Form von scharf abgegrenzten Individuen, sondern als Populationen von Blättern, Stengeln, Wurzeln und Früchten modelliert.

3.2 HERBIVORE

Herbivore konsumieren pflanzliche Biomasse und sind in Agroökosystemen meist poikilotherme Organismen. Ihre Entwicklung unterliegt deshalb ebenfalls stark dem Einfluss der Temperatur. Ausgehend von einer populationsdynamischen Wechselbeziehung zwischen den Pflanzenpopulationen und den Herbivoren können unter Berücksichtigung von gesättigten Funktionalen Reaktionen (Functional Response) (z.B. Holling's disk equation oder analog zum Michaelis-Menten Ansatz) die Entwicklung und Rolle der Herbivoren modelliert werden. Um den Aufwand auf einem erträglichen Niveau zu belassen, sollen die Herbivoren als funktionelle Gruppe im Ökosystem, und nicht auf dem Artniveau modelliert werden. Es ist allerdings auch möglich, sich auf eine Art, z.B. den Maiszünsler *Ostrinia nubilalis* HÜBNER (*Lep.*, *Pyralidae*), der die Maispflanzen merklich zu beeinflussen vermag, zu konzentrieren.

3.3 KARNIVORE

Ähnlich zu den Herbivoren können die Karnivoren modelliert werden. Allerdings sollen hier nicht omnivore, sondern lediglich echt karnivore Organismen modelliert werden, so dass eine eindeutige Nahrungskette entsteht. Um den Aufwand möglichst niedrig zu halten, sollen die Karnivoren lediglich als funktionelle Gruppe im Ökosystem modelliert werden. Es wäre allerdings auch denkbar, sich auf eine Art, z.B. die Erzwespe *Trichogramma evanescens* (Hym., *Calcidae*), die den Maiszünsler befällt, zu konzentrieren.

3.4 DESTRUENTEN

Die Destruenten sollen ebenfalls als eine funktionale Gruppe modelliert werden. Sie bestehen zu einem wesentlichen Teil aus schnell- und kurzlebigen Organismen (z.B. Mikroorganismen), die verhältnismässig rasch auf ein verändertes Nahrungsangebot reagieren. Meist sind diese Organismen limitiert durch den Kohlenstoffanteil im anfallenden Detritus. Ihre Rolle im Agroökosystem ist jedoch besonders für die Erhaltung der Bodenfruchtbarkeit und die Mineralisierung von Nährstoffen, insbesondere Stickstoff N, wichtig.

3.5 WITTERUNG

Die Reaktion eines Ökosystems auf veränderte Klimaverhältnisse kann relativ exakt anhand von Schlüsselprozessen wie der Primärproduktion untersucht werden. Hierzu ist es allerdings notwendig, nicht das Klima, sondern die Witterung im Tagesverlauf zu kennen, da ihr bei biologischen Prozessen eine entscheidende Bedeutung zukommt, die durch das Klima allein nie erfasst werden kann. Da die Witterung als zufälliger Schwankungsprozess um langjährige Mittelwerte verstanden werden kann, soll die Witterung im Tagesverlauf, insbesondere der Temperaturverlauf in seinen Tagesextremen, für die Zukunft prognostiziert (Treibhauseffekt) und für die Gegenwart aus Messungen (SMA) ermittelt werden.

3.6 TREIBHAUSKLIMA

Die Klimaforschung sagt uns für das nächste Jahrhundert eine erhöhte Durchschnittstemperatur voraus (z.B. Schneider, 1989). Das erwartete Treibhausklima kann allerdings durch verschieden Massnahmen der Menschheit, z.B. Reduktion der CO₂-Emissionen, und via unterschiedlicher Annahmen über das Verhalten des globalen Kohlenstoffkreislaufes etwas modifiziert werden. Unter Konzentration auf die Durchschnittstemperatur besteht die Aufgabe darin, ein Zusammenhang zwischen verschiedenen Zukunftsszenarien bezüglich Treibhausgasemissionen wie CO₂, aber auch Fluorchlorkohlenwasserstoffen und Methan zu entwerfen und daraus eine Durchschnittstemperatur für das Jahr 2050 in der Schweiz zu prognostizieren. Besonderer Wert ist bei diesem Unterthema auf die klare Darstellung der gemachten Annahmen und eine Abschätzung ihrer relativen Bedeutung zu legen.

4 Zum Vorgehen und Arbeitsplan

Weitere Angaben, Details und Unterlagen werden später je nach Bedarf durch die Assistenten zur Verfügung gestellt. Bitte Unterlagen aus der Vorlesung Systemanalyse I Teil I A. Fischlin, (Allgemeine Systemtheorie) (WS 88/89) beziehen (s. z.B. 1.1.5 Vorgehen bei der Systemanalyse, s. a. S. 10 - 14: insbesondere Relationengraph, Top-down Modellierung). Arbeitsplan zweite Semesterhälfte SS 89 (Di 15h00 - 17h00 meist im ML D28):

30. Mai Vorstellung Fallstudie «Terrestrisches Ökosystem»; Gruppenarbeit:
 - Schriftlicher Formulierung der Teilaufgabe der Arbeitsgruppe
 - Qualitativer Modellentwurf (Relationengraphen, Gleichungsentwürfe)
 Kennenlernen Ausschaumatrix («Looking Outward Matrix»)

6. Juni Gemeinsame Formulierung der Ausschau matrix zwecks
Koordination
Erste Modellierungsarbeiten an den Untermodellen in Gruppenarbeit
20. Juni Einarbeitung in die Simulationstechnik mit Hilfe von ModelWorks.
Implementierung einfacher Testgleichungen aus dem Untermodell
(Arbeit an den Macintoshrechnern im NO F34.1 und HG E27).
27. Juni Erste Modellierungsarbeiten an den Untermodellen in Gruppenarbeit
Implementierung der Untermodelle mittels ModelWorks.
Verfeinerungen.
4. Juli Verfassung des Schlussberichtes (Text, Modul). Vorbereitung
Synthese
11. Juli Vorstellung des Gesamtmodelles. Schlussdiskussion

5 Übersicht Gesamtmodell

Das Gesamtmodell besteht aus den Untermodellen: Primärproduzenten, Herbivoren, Karnivoren, Destruenten, Witterung und Treibhausklima. Dabei kommt dem Treibhausklimamodell eine besondere Rolle zu. Es hängt von keinem anderen Untermodell ab, erzeugt aber eine wichtige Eingangsgröße für das Witterungsmodell. Da alle anderen Untermodelle von der Witterung beeinflusst werden, werden indirekt alle Untermodelle von dem Untermodell Treibhausklima beeinflusst (Fig. 2).

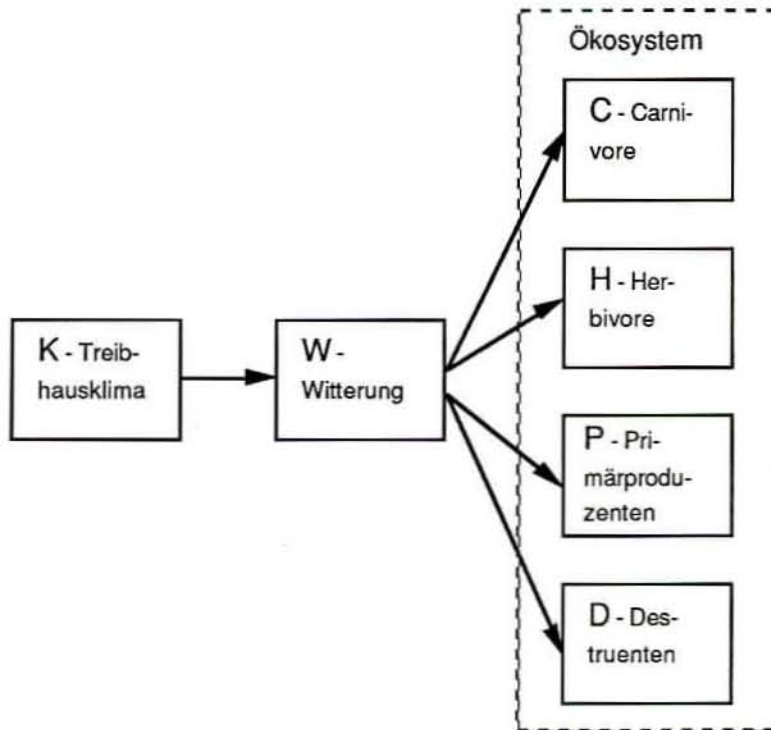


Fig. 2: Klima und Witterungseinflüsse auf das terrestrische Ökosystem.

Mit Ausnahme des Treibhausklimas sind die Untermodelle lediglich innerhalb einem Jahresablauf mit vertretbarem Aufwand zu simulieren. Simuliert man bloss ein einzelnes Jahr, so reduziert sich die Aufgabe des Untermodelles Treibhausklima auf die Erzeugung eines Jahresdurchschnittes für die globale Lufttemperatur. Um diesen Parameter aber trotzdem als ein Resultat einer in Wirklichkeit stattfindenden, langjährigen Entwicklung verstehen zu können, muss für das Untermodell Treibhausklima ein spezieller Zeitmasstab angesetzt werden. Es werden deshalb zwei Arten der Sicht auf das Gesamtmodell unterschieden:

- Langjährige Entwicklung des globalen Klimas (Treibhauseffekt), z.B. 1960 bis 2050
- Verhalten des Agroökosystems innerhalb einer Vegetationsperiode beeinflusst durch das globale Klima für zwei Jahre: 1988 und 2050

Man beachte, dass aber selbst im zweiten Fall das Treibhausuntermodell für eine mehrjährige Periode, nämlich 1960 bis zum gewünschten Jahr, zu Beginn einer eigentlichen Simulation (*Initial*) separat integriert wird. Das erlaubt dann aus der erhaltenen Lösung des Untermodelles Treibhausklima das momentane Jahresmittel der

Temperatur zu berechnen. Dieses Modell des Treibhausklimas ist zeitdiskret, mit einem Zeitschritt von einem Jahr.

Anschließend wird durch das Untermodell Witterung ein spezifischer Temperaturverlauf erzeugt. Dieser Witterungsgenerator basiert auf einem Zufallszahlengenerator und kann für jeden Tag den gesamten Temperaturverlauf, bestimmt durch seine Extremwerte (Maxima und Minima), simulieren. Das Witterungsmodell ist zeitdiskret mit einem Zeitschritt von einem Tag.

Die Untermodelle Primärproduktion, Herbivore, Karnivore und Destruenten enthalten als wichtige Zustandsvariablen die jeweilige Biomasse (Trockensubstanz). Sie sind alle zeitkontinuierlich und als ein System von gekoppelten Differentialgleichungen formuliert. Die jeweiligen Wachstumsraten sind temperaturabhängig und gehorchen alle den folgenden Zusammenhängen (Fig. 3):

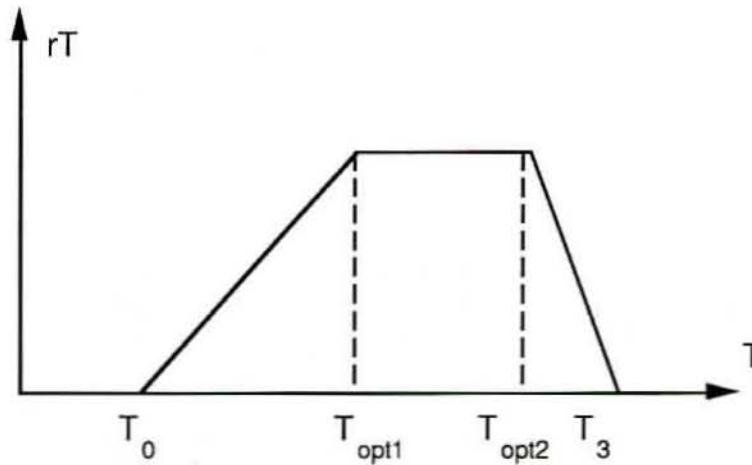


Fig. 3: Abhängigkeit der Wachstumsraten der Primärproduzenten, Herbivoren und Karnivoren von der Temperatur. rT - relative Wachstumsgeschwindigkeit; T - Temperatur. T_0 und T_3 der untere bzw. obere Entwicklungsnullpunkt. T_{opt1} und T_{opt2} begrenzen Temperaturbereich des optimalen Wachstums.

Bei Temperaturen unterhalb des unteren Entwicklungsnullpunktes T_0 sowie des oberen Entwicklungsnullpunktes T_3 findet keine Entwicklung statt. Optimales Wachstum findet innerhalb dem Temperaturbereich T_{opt1} und T_{opt2} statt. Im wichtigen Bereich zwischen T_0 und T_{opt1} wurde die für poikilotherme Organismen übliche linear ansteigende Abhängigkeit der Wachstumsgeschwindigkeit von der Temperatur angenommen.

Zwischen den verschiedenen trophischen Stufen finden Stoffflüsse statt. Es wurden allerdings keine vernetzten trophischen Beziehungen, sondern lediglich eine einfache Nahrungskette modelliert (Fig. 4). Die Destruenten hingegen erhalten Detritus von allen drei trophischen Stufen, d.h. den Primärproduzenten, Herbivoren und Karnivoren (Fig. 4).

Konsum wurde immer als eine Funktion der beiden beteiligten trophischen Stufen aufgefasst. Das Anfallen von Detritus hingegen als ein von den Destruenten unabhängiger Prozess.

Die berücksichtigten Stoffkreisläufe sind mit Ausnahme der N-Mineralisierung durch die Destruenten ausschliesslich Biomasseflüsse.

Die Primärproduzenten spielen im ganzen Ökosystem eine zentrale Rolle. Die Photosynthese und der Erhaltungsstoffwechsel der Primärproduzenten wurden berücksichtigt. Allerdings wurden diese Prozesse lediglich als temperaturbeeinflusst und stickstofflimitiert modelliert.

Bei der Modellierung der biotischen Komponenten, wurde der Einfachheit halber von wenigen Modellorganismen ausgegangen. Sie wurden hierbei als Stellvertreter für die jeweilige funktionelle Gruppe aufgefasst. Beispielsweise wurden für die Primärproduzenten einerseits

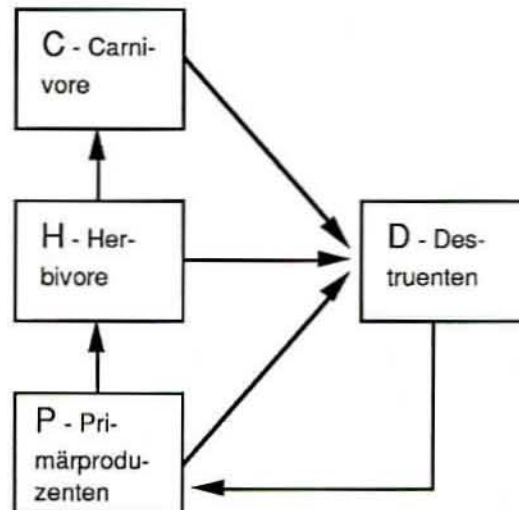


Fig. 4: Stoffflüsse im Agroökosystem zwischen den einzelnen Untermodellen.

Mais- oder Weizenpflanzen angenommen. Die Herbivoren bei Mais wurden durch den Maiszünsler (*O. nubilalis*), für Weizen durch Blattläuse (*Aphidae*) repräsentiert. Die Karnivoren waren beim Maiszünsler *T. evanescens*, bei den Blattläusen durch Marienkäfer (*Coccinellidae*) vertreten.

6 Ausschaumatrix

Die Ausschaumatrix («Looking-Outward Matrix») erlaubt das Zusammenarbeiten von verschiedenen Spezialisten, die für sich in Gruppen, aber auf ein gemeinsames Ziel hin ein Modell erarbeiten. Sie beruht auf dem Prinzip, dass die jeweiligen Spezialisten nicht für sich selbst bestimmen dürfen, was sie den anderen für die Zusammenarbeit anzubieten haben, sondern dass die jeweiligen Gruppen zuerst spezifizieren, was sie von den anderen Spezialisten zur Lösung ihrer Aufgabe benötigen.

Looking Outward Matrix

	Primärproduktion	Herbivore	Karnivore	Destruenten	Witterung	Treibhausklima
Primärproduktion	-----	Pflanzliche Biomasse (Trockengewicht) [kg(TS)/ha] <i>dmPrimProd</i>	-----	Pflanzlicher Detritus [kg(TS)/ha/d] Stickstoffkonsum [kg(N)/ha/d] <i>detrPrimProd, consN</i>	-----	-----
Herbivore	Herbivorenkonsum (Trockensubstanz) [kg(TS)/ha/d] <i>consHerbivores</i>	-----	Herbivoren Biomasse (Trockensubstanz) [kg(TS)/ha] <i>dmHerbivores</i>	Herbivorer Detritus (Trockensubstanz) [kg(TS)/ha/d] <i>detrHerbivores</i>	-----	-----
Karnivore	-----	Karnivorenkonsum (Trockensubstanz) [kg(TS)/ha/d] <i>consCarnivores</i>	-----	Karnivorer Detritus (Trockensubstanz) [kg(TS)/ha/d] <i>detrCarnivores</i>	-----	-----
Destruenten	N-Gehalt [kg(N)/ha] <i>NCont</i>	-----	-----	-----	-----	-----
Witterung	Tagestemperatur Minimum, Maximum [°C] <i>dTmin, dTmax</i>	Tagestemperatur Minimum, Maximum [°C] <i>dTmin, dTmax</i>	Tagestemperatur Minimum, Maximum [°C] <i>dTmin, dTmax</i>	Tagestemperatur Minimum, Maximum [°C] <i>dTmin, dTmax</i>	-----	-----
Treibhausklima	-----	-----	-----	-----	Globale Lufttemperatur Jahresmittel [°C] <i>annualMeanT</i>	-----

Fallstudie Treibhauseffekt und Terrestrisches Ökosystem

kursive Angaben: Variablennamen, wie sie für die Computer-Programmierung des Gesamtmodells verwendet wurden.

7 Methodik der Realisierung

Dabei geht man so vor, dass jede Gruppe zunächst die ihr zugehörige Kolonne in der Ausschaumatrix ausfüllt. Dabei muss sie das Diagonalelement auslassen. Eingetragen werden die auszutauschenden Grössen inklusive deren Einheiten. Damit wird das reibungslose Zusammenfügen der Einzelmodelle gewährleistet. Die zu erzeugenden Ausgangsvariablen der jeweiligen Untermodelle ergeben sich dann durch das Ablesen der entsprechenden Zeile. Beispielsweise ergibt sich für die Gruppe Herbivore (siehe Ausschaumatrix nächste Seite), dass sie den Primärkonsum, die für Karnivore potentiell konsumierbare Biomasse und den herbivoren Detritus mit den Gleichungen des Herbivorenmodelles bestimmen können muss. Wie im Einzelnen das geschieht, ist Sache der Gruppe von Spezialisten, die sich mit den Herbivoren beschäftigt, und ist für das Zusammenfügen belanglos. Letzteres gilt natürlich für das Verhalten des Gesamtmodelles nicht. Die Ausschaumatrix bestimmt also lediglich die Ein- bzw. Ausgangsvariablen, die zur Koppelung der Untermodelle benötigt werden.

Einmal bestimmt, kann die Ausschaumatrix direkt als Grundlage zum Entwurf von Definitionsmodulen eines ModelWorks Modell Programmes (Fischlin & Ulrich, 1989) verwendet werden.

Es wurde als erstes ein Gesamtmodell mit Hilfe der Ausschaumatrix (s.o.) entworfen. Anschliessend wurden durch Arbeiten in Gruppen die Gleichungen für die jeweiligen Untermodelle separat entwickelt. In einem letzten Schritt sind die einzelnen Untermodelle mit Hilfe von ModelWorks implementiert worden (Fischlin & Ulrich, 1989).

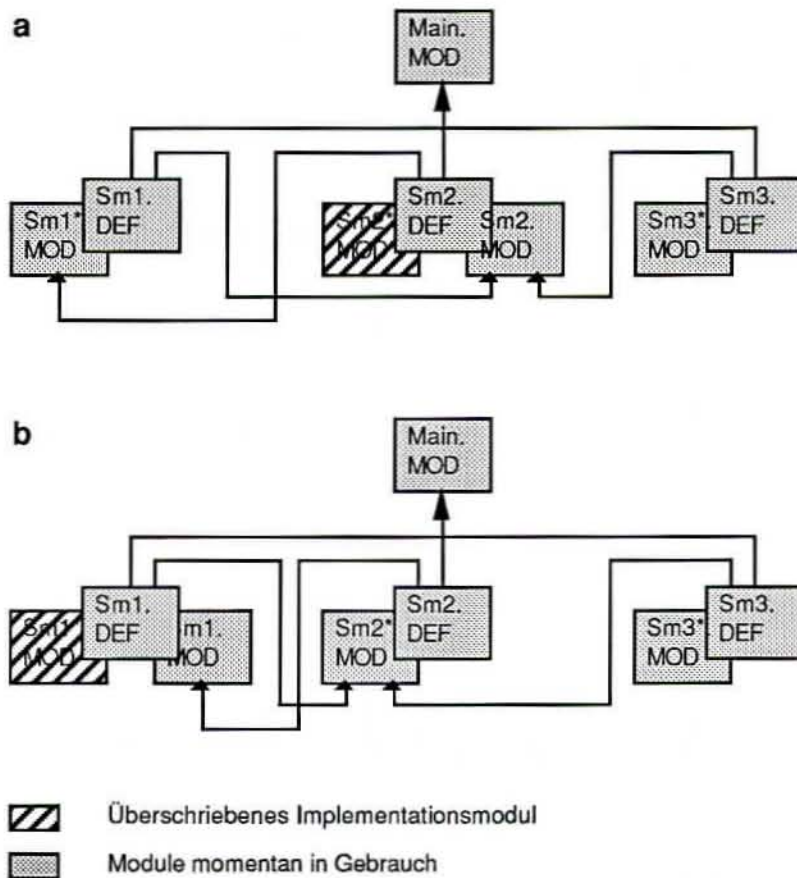


Fig. 5: Situationsbeispiel während der Implementierung der verschiedenen Untermodelle. Für das jeweilige Entwerfen und Austesten eines einzelnen Untermodelles stehen alle anderen benötigten Module durch eine einfache vordefinierte Implementation zur Verfügung, obwohl sich alle anderen Modelle ebenfalls gerade in Bearbeitung befinden. a) Das zweite Sm2 b) Das erste Untermodell Sm1 ist in Bearbeitung. Pfeile entsprechen Importen.

Jedem Untermodell entspricht ein Modula-2 Modul (Wirth, 1985). Es bestehen indirekte, zirkuläre Abhängigkeiten der Untermodelle gemäss der Ausschau matrix (s.a. Fig. 4). Um trotzdem unabhängig von den Parallelentwürfen aller anderen Gruppe jeweils einer bestimmten Gruppe das Implementieren ihres Untermodelles zu ermöglichen, wurden für alle Untermodelle eine provisorische Modellrealisierung zur Verfügung gestellt (xyz*.MOD Implementation mit gleichem Modulbezeichner aber separate Datei). Diese Modellimplementationen enthalten keine dynamischen Modellgleichungen, sondern es wird das Verhalten der Ausgangsvariablen der einzelnen Untermodelle als Funktion der Zeit abgeschätzt, und mit Hilfe von Tabellenfunktionen implementiert. Dadurch stand jeder Gruppe jederzeit ein ausführbares Gesamtmodell zur Verfügung, das das Implementieren und Austesten ihres Untermodelles völlig unabhängig von allenfalls schon oder noch nicht vorliegenden Resultaten der anderen Gruppen erlaubte (xyz.MOD Implementation mit gleichem Modulbezeichner aber neue, separate Datei). Die jeweils benötigten Eingangsvariablen wurden durch die Tabellenfunktionen erzeugt, und die neue, eigentliche Modellimplementation (xyz.MOD) ersetzte die vordefinierte Implementation (xyz*.MOD) des jeweiligen Untermodells (Fig. 5).

8 Untermodelle

8.1 PRIMÄRPRODUZENTEN

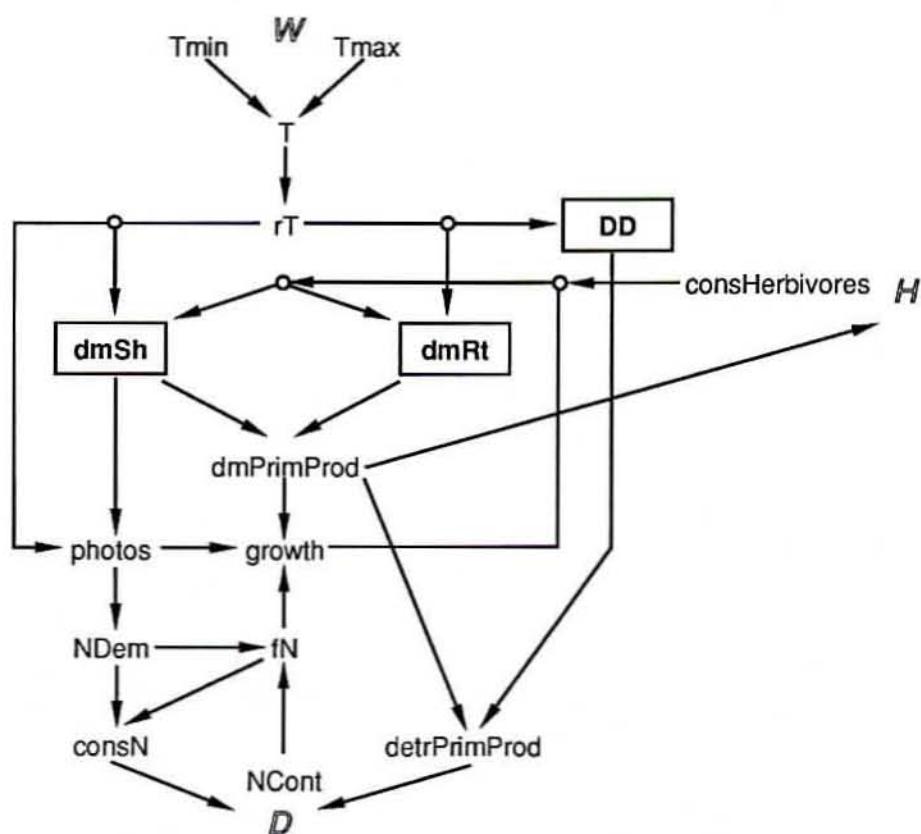


Fig. 6: Relationengraph für das Untermodell PrimProd (Zustandsgrößen eingerahmt, andere Submodelle Konturschrift).

Zustandsvariablen	Einheiten	Anfangswerte	
		Mais	Weizen
dmSh (weight of shoots)	kg(TS)/ha	100	100
dmRt (weight of roots)	kg(TS)/ha	50	50
DD (degreedays)	°C-d	0	0

Tagesgrade:

$$\frac{dDD}{dt} = rT \cdot (Topt_1 - T_0)$$

Temperaturabhängigkeit (siehe Fig. 3):

$$rT = \left\{ \begin{array}{ll} \frac{T - T_0}{T_{opt1} - T_0} & T_0 < T \leq T_{opt1} \\ 1.0 & T_{opt1} < T \leq T_{opt2} \\ \frac{T_3 - T}{T_3 - T_{opt2}} & T_{opt2} < T \leq T_3 \end{array} \right\} \quad 0 \leq rT \leq 1$$

Photosynthese:

$$photos = g_{phot} \cdot rT \cdot (T_{opt1} - T_0) \cdot (1 - e^{-ext \cdot sla \cdot dmSh})$$

Gesamtmasse:

$$dmPrimProd = (dmSh + dmRt)$$

Erhaltung:

$$maint = dmPrimProd \cdot r_{maint}$$

Absterberate:

$$detrPrimProd = dmPrimProd \cdot dr$$

Wachstumslimitierung wegen N-Konsum:

$$NDem = photos \cdot N_{ConcInPl}$$

$$fN = \frac{\text{Min}(NDem, N_{Cont})}{NDem} \quad 0 \leq fN \leq 1$$

Wachstum:

$$growth = (fN \cdot photos - maint) \cdot cvf$$

$$\frac{dDmSh}{dt} = (growth - consH - detrPrimProd) \cdot fsh$$

$$\frac{dDmRt}{dt} = (growth - consH - detrPrimProd) \cdot (1 - fsh)$$

Parameter	Einheiten	Werte	
		Mais	Weizen
cvf	kg(TS)/kg(TS)	0.7	0.7
fsh	%/100	0.6	0.6

Parameter (Forts.)	Einheiten	Werte	
		Mais	Weizen
d	kg(TS)/kg(TS)/d	0.005	0.005
ext	%/100	0.8	0.8
$gphot$	kg(TS)/(°C·d)	40.0	30.0
$NConclnPl$	%/100	0.01	0.01
$rmaint$	kg(TS)/kg(TS)/d	0.015	0.015
sla	ha/kg(TS)	0.0025	0.002
T_0	°C	6.0	5.0
T_3	°C	35.0	30.0
$Topt_1$	°C	20.0	15.0
$Topt_2$	°C	25.0	20.0

8.2 HERBIVORE

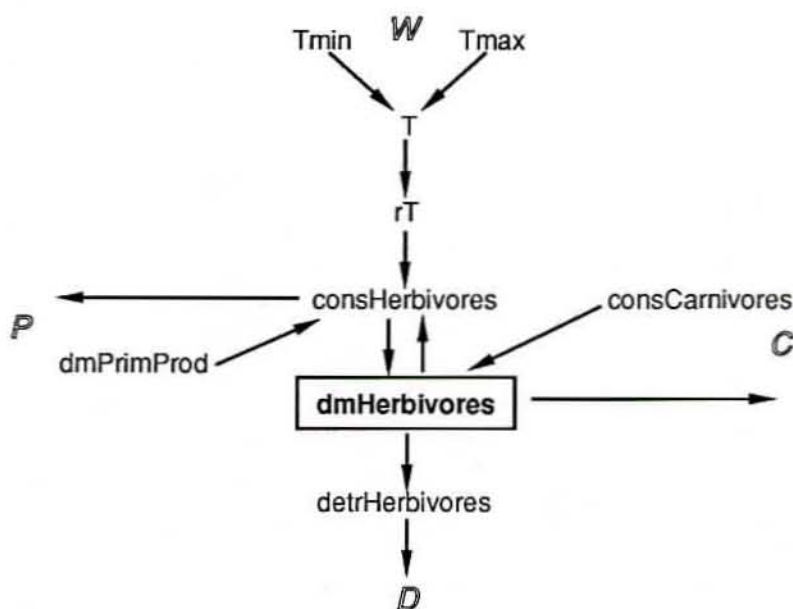


Fig. 7: Relationengraph für das Untermodell Herbivoren (Zustandsgrößen eingerahmt, andere Submodelle Konturschrift).

Zustandsvariablen	Einheiten	Anfangswerte	
		Maiszünsler	Blattläuse
dmHerbivores (dry matter herbivores)	kg(TS)/ha	100	100

Temperaturabhängigkeit (siehe auch Fig. 3):

rT wie bei Primärproduktion, jedoch andere Parameter

Funktionale Reaktion (nach Michaelis-Menten):

$$f_{1B} = f_{max} \cdot \frac{dm_{PrimProd}}{dm_{PrimProd} + \beta}$$

$$cons_{Herbivores} = \text{Min} (dm_{Herbivores} \cdot f_{1B} \cdot r_T, dm_{PrimProd})$$

Absterberate:

$$detr_{Herbivores} = dm_{Herbivores} \cdot dr$$

Wachstumsrate:

$$\frac{dDm_{Herbivores}}{dt} = cvf \cdot cons_{Herbivores} - detr_{Herbivores} - cons_{Herbivores}$$

Parameter	Einheiten	Werte	
		Maiszünsler	Blattläuse
β	kg(TS)/ha	100.0	100.0
cvf	kg(TS)/kg(TS)/d	0.7	0.7
dr	kg(TS)/kg(TS)/d	0.02	0.02
f_{max}	kg(TS)/ha	0.11	0.11
T_0	°C	10.0	4.0
T_3	°C	35.0	30.0
T_{opt1}	°C	20.0	18.0
T_{opt2}	°C	25.0	22.0

Zustandsvariablen	Einheiten	Anfangswerte	
		Trichogramma	Coccinelliden
$dm_{Carnivores}$ (dry matter carnivores)	kg(TS)/ha	10	10

Temperaturabhängigkeit (siehe auch Fig. 3):

r_T wie bei Primärproduktion, jedoch andere Parameter

Funktionale Reaktion (geknickte Lineare):

$$consume = \text{Min} (dm_{Carnivores} \cdot dm_{Herbivores} \cdot a, dm_{Carnivores} \cdot max_{consume})$$

$$cons_{Carnivores} = \text{Min} (r_T \cdot consume, dm_{Herbivores})$$

Absterberate:

$$detr_{Carnivores} = dm_{Carnivores} \cdot dr$$

Wachstumsrate:

$$\frac{dDmCarnivores}{dt} = cvf \cdot consCarnivores - detrCarnivores$$

8.3 KARNIVORE

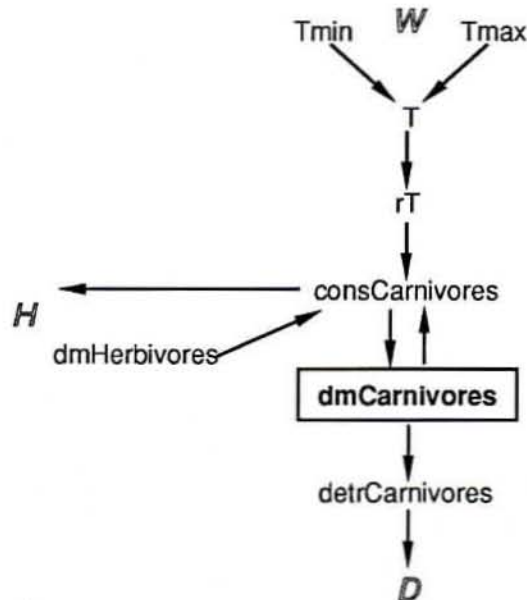


Fig. 8: Relationengraph für das Untermodell Herbivoren (Zustandsgrößen eingrahmt, andere Submodelle Konturschrift).

Parameter	Einheiten	Werte	
		Trichogramma	Coccinelliden
a	kg(TS)/kg(TS)/d	0.005	0.005
cvf	kg(TS)/kg(TS)	0.7	0.7
d	kg(TS)/kg(TS)/d	0.02	0.02
$maxConsume$	kg(TS)/kg(TS)/d	0.2	0.2
T_0	°C	12.0	8.0
T_3	°C	35.0	35.0
$Topt1$	°C	25.0	22.0
$Topt2$	°C	30.0	28.0

8.4 DESTRUENTEN

Zustandsvariablen	Einheiten	Anfangswerte	
		Mais	Weizen
dmDetr (dry matter detritus)	kg(TS)/ha	120000	120000
N (total amount of N)	kg(N)/ha	3000	3000

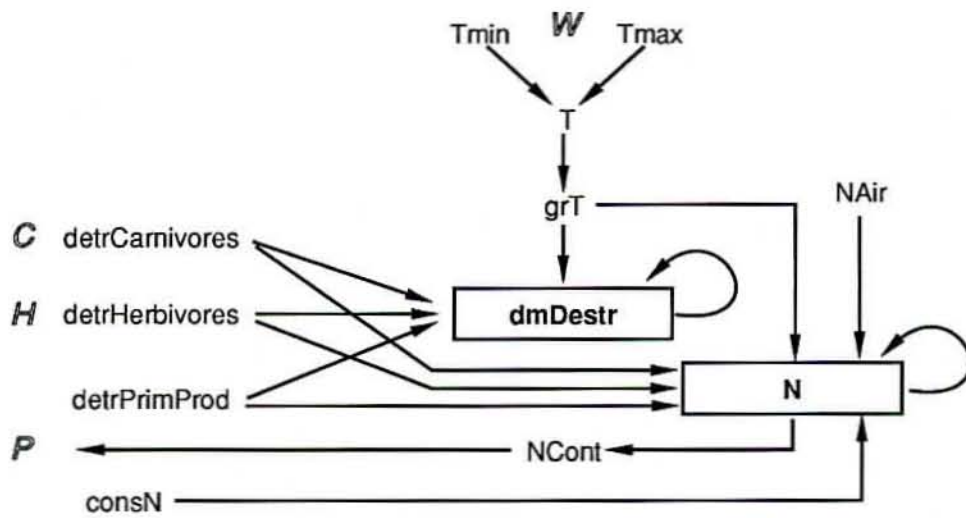


Fig. 9: Relationengraph für das Untermodell Herbivoren (Zustandsgrößen eingerahmt, andere Submodelle Konturschrift).

Temperaturabhängigkeit:

$$grT = \text{Max} \left(\frac{b_1 + b_2 \cdot T + b_3 \cdot T^2}{grT_{\text{max}}}, 0 \right)$$

Anfall von neuem Detritus (inklusive Detritus aus sich selber):

$$C_D = DmDestr \cdot dr \cdot grT \cdot (1 - f_{lost})$$

$$C_{suppl} = C_D + \sum_{i=P}^C detr_i \cdot CConc_i$$

$$N_D = N \cdot dr \cdot grT \cdot (1 - f_{lost})$$

$$N_{suppl} = N_D + N_{AIR} + \sum_{i=P}^C detr_i \cdot NConc_i$$

Wachstumsraten:

$$\frac{dDmDestr}{dt} = grT \cdot grEff \cdot C_{suppl} - C_D$$

$$\frac{dN}{dt} = grT \cdot N_{suppl} - N_D - consN$$

Parameter	Einheiten	Werte	
		Mais	Weizen
<i>b1</i>	-	-1.16	-1.16
<i>b2</i>	1/°C	0.1907	0.1907
<i>b3</i>	1/(°C) ²	-0.0026	-0.0026
<i>CConcCa</i>	kg(TS)/kg(TS)	0.89	0.89
<i>CConcHe</i>	kg(TS)/kg(TS)	0.89	0.89
<i>CConcPp</i>	kg(TS)/kg(TS)	0.98	0.98
<i>d</i>	%/100/d	0.002	0.002
<i>flost</i>	%/100	0.3	0.3
<i>grEff</i>	kg(TS)/kg(TS)	0.1	0.1
<i>minr</i>	kg(N)/kg(N)	0.003	0.003
<i>NAir</i>	kg(N)/d	0.1	0.1
<i>NConcCa</i>	kg(N)/kg(TS)	0.10	0.10
<i>NConcHe</i>	kg(N)/kg(TS)	0.10	0.10
<i>NConcPp</i>	kg(N)/kg(TS)	0.01	0.01

8.5 WITTERUNG

Annahmen: Alle statistischen Grössen sind mit Ausnahme der Erwartungswerte als konstant angesehen worden. Der Treibhauseffekt erhöht demnach lediglich den Mittelwert der Lufttemperatur. Der Jahresgang der Temperatur wurde ebenfalls als gleichbleibend angenommen und basiert auf den langjährigen Monatsmitteln, die um den Anstieg der globalen Jahrestemperatur bei erfolgtem Treibhauseffekt erhöht werden.

Daten: Es wurden langjährige Mittel verschiedenster Parameter der Lufttemperatur wie Monatsmittel $T_m(i)^*$, Tagesmittel T_d^* , Tagesamplitude a_d^* ($a_d = T_{max} - T_{min}$) und die jeweiligen Standardabweichungen sT_m^* , sT_d^* , sowie s_{ad}^* verwendet. Die Daten stammen alle von der Messtation SMA in Zürich (Schüepp 1961) und sind:

T_a^*	8.2 °C	Jahresmittel											
<i>i</i>	Jan	Feb	Mar	Apr	Mai	Jun	Jul	Aug	Sep	Okt	Nov	Dez	
$T_m(i)^*$	-1.0	0.2	4.2	8.0	12.5	15.5	17.2	16.6	13.5	8.4	3.3	0.2 °C	

Langjährige Jahresmittelwerte:

Symbol	Wert	Bezeichner	Bedeutung
sT_m^*	2.769	mTstdevStar	Standardabw. Monatsmittel Lufttemperatur
sT_d^*	1.98	dTstdevStar	Standardabw. Tagesmittel Lufttemperatur
s_{ad}^*	4.35	dTampstdevStar	Standardabw. Tagesamplitude Lufttemperatur
a_d^*	9.08	dTampStar	Mittlere Amplitude Tagesextrema Lufttemperatur
<i>m</i>	0.6	mem	Koeffizient Einfluss Vortagestemperatur

Zustandsvariablen	Einheiten	Anfangswerte
T_d	°C	-1.0

Gleichungen:Zustandsvariable T_d (Tagesmittel) und Differenzgleichung (Zeitschritt 1d):

$$T_d' = N(T_m(i)^*, s_{T_m}^*)$$

$$T_d(k+1) = m \cdot T_d(k) + N((1-m) \cdot T_d', (1+m) \cdot s_{T_d}^*)$$

$$a(k) = N(a_d^*, s_{a_d}^*)$$

wobei $N(\mu, \sigma)$ ist eine Realisierung einer normalverteilten Zufallsvariable mit dem Erwartungswert μ (arithmetisches Mittel) und der Standardabweichung der Einzelwerte (Streuung) σ .

$$T_{\min} = T_d(k) - 0.5 \cdot a(k)$$

$$T_{\max} = T_d(k) + 0.5 \cdot a(k)$$

8.6 TREIBHAUSKLIMA

Szenarien:

- L Lineares Wachstum - Extrapolation entlang momentaner Steigung (1989) $\Delta = -0.11$ /a
- E Exponentielles Wachstum - Extrapolation aus Entwicklung präindustrieller Zeit (~1850) bis Gegenwart (1989) $t_s = 2000$ a, $r_{\max} = 0.023$ %/a
- F «Kalte Fusion» Rückgang auf Istzustand nach $t_s = 40$ Jahren bei $r_{\max} = 0.023$ %/a

Gleichungen:

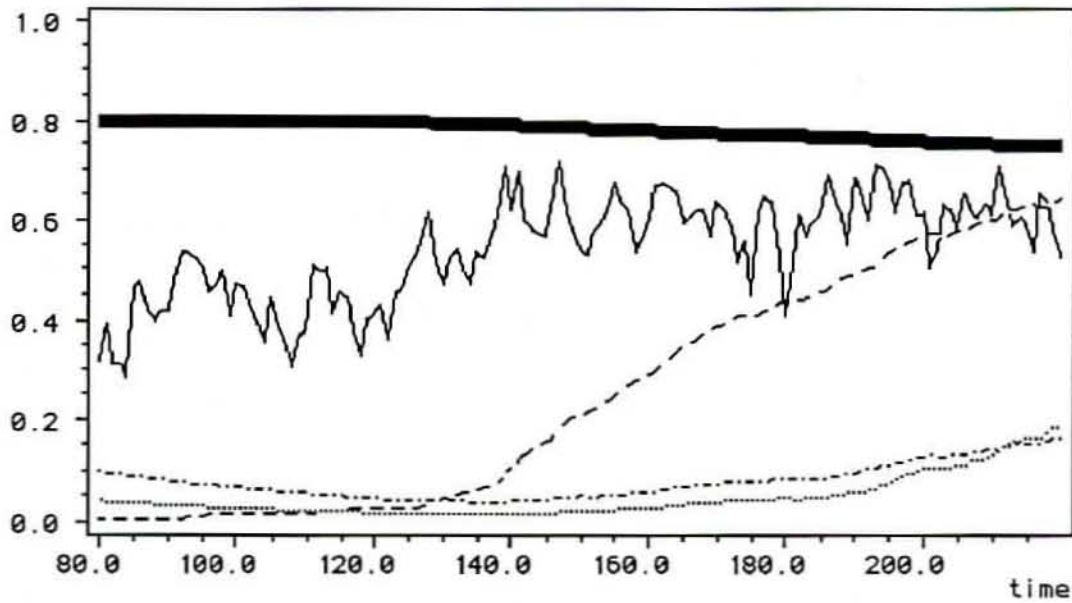
$$b'(t+1) = (1+r(t)) b(t) + \Delta \quad \text{mit} \quad r(t) = r_{\max} (t_s - (t-t_0))/t_s$$

$$c(t) = c(1850) + \gamma b(t)$$

$$T(t) = T(1960) + (c(t) - c(1960)) \tau$$

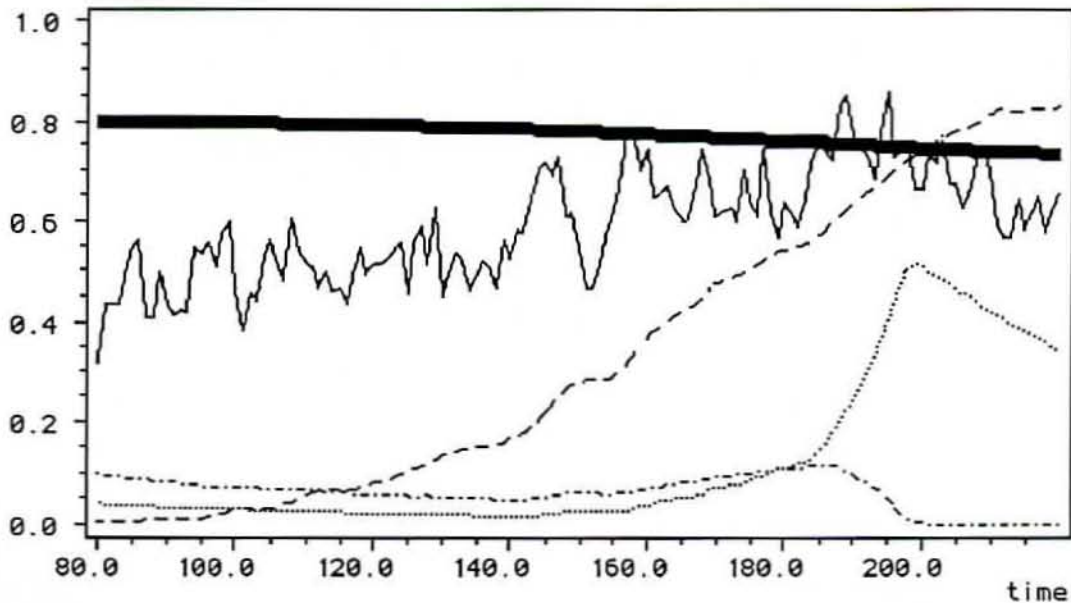
Zustandsvariablen		Einheiten	Anfangswerte
b	Verbrauch fossiler Brennstoffe	Gt C	5.5 (1988)
Hilfsvariablen		Einheiten	Anfangswerte
c	CO ₂ -Gehalt Atmosphäre	ppm	310 (1960)
T	Globale Temperatur Jahresmittel	°C	15.0 (1960)
Parameter		Einheiten	Werte
r	Maximaler Konsumzuwachs Fossile Brennst.	%/100a	0.023 (s. Szenarien) 0.005 0.053
γ	Anstieg CO ₂ in Atmosphäre per Emmiss.	ppm/Gt	10.0
Δ	Konsuminkrement	Gt/a	0.075 (L) 0.0 (E) -0.11 (F)
τ	Temperaturanstieg pro CO ₂ -Gehalt	°C/ppm	0.012

9 Resultate



Plotted curves:	Minimum	Maximum
— dTmean	-20.000	40.000
- - - dmPrimProd	0.000	25000.000
· · · · dmHerbivores	0.000	1000.000
· · · · · dmCarnivores	0.000	250.000
● dmDestr	0.000	150000.00

Fig. 10: Simulationsergebnisse für das Agroökosystem Mais für das Jahr 1988 (Wetter simuliert).



Plotted curves:	Minimum	Maximum
— dTmean	-20.000	40.000
- - - dmPrimProd	0.000	25000.000
· · · · dmHerbivores	0.000	1000.000
· · · · · dmCarnivores	0.000	250.000
● dmDestr	0.000	150000.00

Fig. 11: Simulationsergebnisse für das Agroökosystem Mais für das Jahr 2050 (Szenario E1).

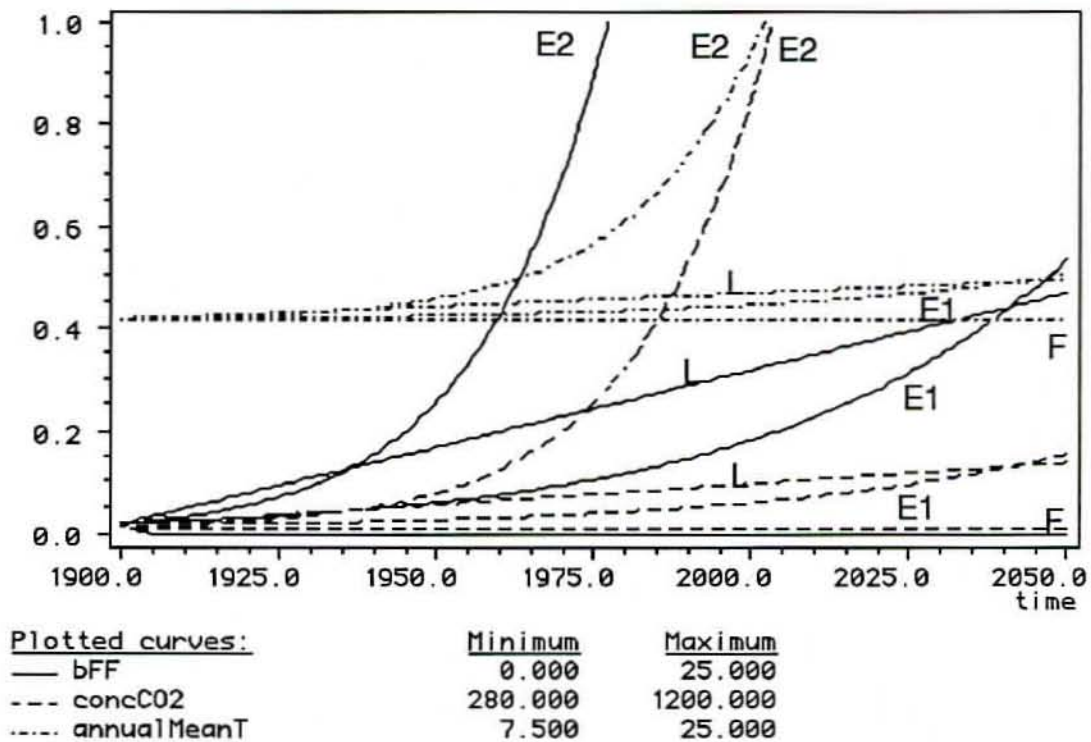


Fig. 12: Simulationsergebnisse des Klimamodelles für verschiedene Szenarien. Legende: L = lineare Zunahme ($r=0$, $\Delta=0.07$, $ts=2000$); E1 = exponentielle Zunahme wie nach Ölkrise ($r=0.023$, $\Delta=0$, $ts=2000$); E2 = exponentielle Zunahme vor Ölkrise ($r=0.053$, $\Delta=0$, $ts=2000$); F = "Kalte Fusion" ($r=0.02$, $\Delta=-0.11$, $ts=40$).

10 Literatur

- ELLENBERG, H. (ED.), 1973. *Ökosystemforschung*. Springer, Berlin a.o., 280pp.
- FISCHLIN, A. & ULRICH, M., 1989. *ModelWorks. Manual for an interactive Modula-2 modelling and simulation environment*. Systems Ecology, Swiss Federal Institute of Technology Zürich, Switzerland, 120pp.
- PENNING DE VRIES, F.W.T. & VAN LAAR, H.H. (EDS.), 1982. *Simulation of plant growth and crop production*. Simulation Monographs, Centre for Agricultural Publishing and Documentation, Wageningen, 308pp.
- SCHNEIDER, S.H., 1989. *The greenhouse effect: science and policy*. Science, **243**: 771-81.
- SCHÜEPP, M., 1967. *Klimatologie der Schweiz C. Lufttemperatur (3. und 4. Teil)*. Beiheft zu den Annalen der Schweizerischen Meteorologischen Anstalt (SMA) (Jahrgang 1966), Fotorotar AG, Zürich, p. C/70, p. C/89, p. C/224 und p. C/225, p. C/232
- WIRTH, N., 1985. *Programming in Modula-2, Third, Corrected Edition*. Springer-Verlag, Berlin a.o., 202pp.

Anhang (Programmlistings)

A GESAMTMODELL - PROGRAMMODUL
(HAUPTPROGRAMM) TERRECOSYS

MODULE TerrEcoSystem ;

Version written for:
 MacMETH V2.5.1 (1-Pass Modula-2 implementation)
 Dialog Machine V1.0 (User interface)
 ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Main model
 Imports several submodels and has been constructed
 to study the impact of the greenhouse effect on a
 terrestrial ecosystem in the year 2050

It is part of the case study model developed
 in the course "Systemanalyse II (Naturwissen-
 schaftliche Anwendungen) Teil A. Fischlin, SS 85".
 The synthesis of the whole model is made by
 means of the "looking outward matrix".

List of imported submodels:

K - GreenhClim	Treibhausklima
W - Weather	Witterung
P - PrimProd	Primärproduzenten
H - Herbivores	Herbivore
C - Carnivores	Karnivore
D - Destruents	Destruenten

Remarks: version implemented by Systems Ecology Group

Remarks: this is the ModelWorks supermodel making calls
to all submodels' model declaration procedures

Implementation and Revisions:

Author	Date	Description
--------	------	-------------

dg	23/05/89	Implementation of module-structure
af	30/06/89	Weather now declared second (right after green-house climate) to guarantee correct inputs (Tmin, Tmax) even if used to compute outputs of the subsequent submodels. SetXYZMonitoring procedures added

```
FROM DMStrings IMPORT AssignString;
FROM DMEnterForms IMPORT FormFrame, WriteLabel, RadioButtonID,
  DefineRadioButtonSet, RadioButton, CheckBox, IntField, DeflUse,
  UseEntryForm;
FROM DMMenus IMPORT Menu, Command, AccessStatus, Marking,
  Separator, InstallAbout, InstallMenu, InstallCommand,
  InstallAliasChar;
```

```
FROM SimBase IMPORT
  SetSimTime, SetMonInterval, SetIntegrationStep,
  DeclExperiment ;
FROM SimMaster IMPORT RunSimMaster, SimRun ;
```

```
IMPORT GreenhClim ;
IMPORT Weather ;
IMPORT PrimProd ;
IMPORT Herbivores ;
IMPORT Carnivores ;
IMPORT Destruents ;
```

```
CONST
  KName = " K - Klima";
  WName = " W - Witterung";
  PName = " P - Primärproduzenten";
  HName = " H - Herbivore";
  CName = " C - Karnivore";
  DName = " D - Destruenten";
```

```
VAR
  useManyOrlYear: RadioButtonID;
  useForManyYears, useForlYear: RadioButtonID;
  useGreenhClimForManyYears,
  useWeather,
  usePrimProd,
  useHerbivores,
  useCarnivores,
  useDestruents,
  notCancelled: BOOLEAN;

  maxRunNo: REAL;
```

```
PROCEDURE ChooseModels;
  CONST lm = 5;
  VAR bf: FormFrame; ok: BOOLEAN; cl: INTEGER;
  BEGIN
    cl := 2;
```

```
  WriteLabel(cl, lm, "Wähle Modell:"); INC(cl);
  DefineRadioButtonSet(useManyOrlYear);
  RadioButton(useForManyYears, cl, lm+1,
    "Langfristige Klimaentwicklung"); INC(cl); INC(cl);
  RadioButton(useForlYear, cl, lm+1,
    "Jahresverlauf bei gegebenem Klima"); INC(cl);
  IF useGreenhClimForManyYears THEN
    useManyOrlYear := useForManyYears;
  ELSE
    useManyOrlYear := useForlYear;
  END(*IF*);
  CheckBox(cl, lm+3, WName, useWeather); INC(cl);
  CheckBox(cl, lm+3, PName, usePrimProd); INC(cl);
  CheckBox(cl, lm+3, HName, useHerbivores); INC(cl);
  CheckBox(cl, lm+3, CName, useCarnivores); INC(cl);
  CheckBox(cl, lm+3, DName, useDestruents); INC(cl);
  INC(cl);
  bf.x := 0; bf.y := -1;
  bf.lines := cl; bf.columns := 60;
  UseEntryForm(bf, notCancelled);
  IF notCancelled THEN
    IF useManyOrlYear = useForManyYears THEN
      useGreenhClimForManyYears := TRUE;
    ELSEIF useManyOrlYear = useForlYear THEN
      useGreenhClimForManyYears := FALSE;
    END(*IF*);
  END(*IF*);
END ChooseModels;
```

```
CONST
  K = 0; W = 1; P = 2; H = 3; C = 4; D = 5;
  minLev = 0; maxLev = 2;
```

```
VAR
  terEcoSysMenu: Menu; setESKCmd, setMonLevCmd: Command;
  bts: ARRAY [K..D] OF RECORD
    rbSet: RadioButtonID;
    rb: ARRAY [minLev..maxLev] OF RadioButtonID;
    lev: [minLev..maxLev];
    nam: ARRAY [0..80] OF CHAR;
  END(*RECORD*);
  curEcsk: PrimProd.EcoSysKind;
```

PROCEDURE SetEcoSysKind(ecsk: PrimProd.EcoSysKind);

```
BEGIN
  IF NOT useGreenhClimForManyYears THEN
    IF usePrimProd THEN
      PrimProd.SetPPEcoSysKind(ecsk);
    END(*IF*);
    IF useHerbivores THEN
      Herbivores.SetHEcoSysKind(ecsk);
    END(*IF*);
    IF useCarnivores THEN
      Carnivores.SetCEcoSysKind(ecsk);
    END(*IF*);
    curEcsk := ecsk;
  END(*IF*);
END SetEcoSysKind;
```

PROCEDURE ChooseEcoSysKind;

```
CONST lm = 5;
VAR bf: FormFrame; cl: INTEGER; str: ARRAY [0..1] OF CHAR;
  rbSet, corn, wheat: RadioButtonID;
  BEGIN
    cl := 2;
    WriteLabel(cl, lm, "Wähle Agroökosystem:"); INC(cl);
    DefineRadioButtonSet(rbSet);
    RadioButton(corn, cl, lm+1, "Mais"); INC(cl);
    RadioButton(wheat, cl, lm+1, "Weizen"); INC(cl);
    IF curEcsk = PrimProd.Wheat THEN rbSet := wheat;
  END(*IF*);
  bf.x := 0; bf.y := -1;
  bf.lines := cl; bf.columns := 30;
  UseEntryForm(bf, notCancelled);
  IF notCancelled THEN
    IF rbSet = corn THEN
      curEcsk := PrimProd.Maize;
    ELSE
      curEcsk := PrimProd.Wheat;
    END(*IF*);
    SetEcoSysKind(curEcsk);
  END(*IF*);
END ChooseEcoSysKind;
```

PROCEDURE SetMonitoring;

```
BEGIN
  IF NOT useGreenhClimForManyYears THEN
    GreenhClim.SetGreenhouseClimateMonitoring(bts[K].lev);
  IF NOT useGreenhClimForManyYears THEN
    IF useWeather THEN
      Weather.SetWeatherMonitoring(bts[W].lev);
    END(*IF*);
    IF usePrimProd THEN
      PrimProd.SetPrimaryProductionMonitoring(bts[P].lev);
    END(*IF*);
    IF useHerbivores THEN
      Herbivores.SetHerbivoresMonitoring(bts[H].lev);
    END(*IF*);
    IF useCarnivores THEN
      Carnivores.SetCarnivoresMonitoring(bts[C].lev);
    END(*IF*);
    IF useDestruents THEN
      Destruents.SetDestruentsMonitoring(bts[D].lev);
    END(*IF*);
  END(*IF*);
END SetMonitoring;
```

PROCEDURE ChooseMonitoring;

```
CONST lm = 5;
VAR i: [K..D]; j: [minLev..maxLev]; cl: INTEGER;
  bf: FormFrame; str: ARRAY [0..1] OF CHAR;
  BEGIN
    cl := 2;
    WriteLabel(cl, lm, "Wähle Monitoringniveau:"); INC(cl);
    FOR j := minLev TO maxLev DO
      str[0] := CHR(j+ORD("0")); str[1] := OC;
```

```

WriteLabel(cl,lm+1 + j*3, str );
END(*FOR*);
INC(cl);
FOR i:= K TO D DO
DefineRadioButtonSet(bts[i].rbSet);
FOR j:= minLev TO maxLev DO
IF j<maxLev THEN
RadioButton( bts[i].rb[j] , cl,lm+1 + j*3,"");
ELSE
RadioButton( bts[i].rb[j] , cl,lm+1 + j*3, bts[i].nam);
END(*IF*);
IF j=bts[i].lev THEN bts[i].rbSet:= bts[i].rb[j] END(*IF*);
END(*FOR*);
INC(cl);
END(*FOR*);
INC(cl);
bf.x:= 0; bf.y:= -1;
bf.lines:= cl; bf.columns:= 44;
UseEntryForm( bf, notCancelled );
IF notCancelled THEN
FOR i:= K TO D DO
FOR j:= minLev TO maxLev DO
IF bts[i].rbSet=bts[i].rb[j] THEN bts[i].lev:= j END;
END(*FOR*);
END(*FOR*);
SetMonitoring;
END(*IF*);
END ChooseMonitoring;

PROCEDURE DoSeveralRuns;
VAR l,maxRunNo: INTEGER;
PROCEDURE HowManyRuns;
CONST lm = 5;
VAR cl: INTEGER; bf: FormFrame;
BEGIN
cl:= 2;
WriteLabel(cl,lm,"Anzahl Läufe");
maxRunNo:= 5;
IntField(cl,lm+30,5,maxRunNo,useAsDeflt,0,MAX(INTEGER));
INC(cl); INC(cl);
bf.x:= 0; bf.y:= -1;
bf.lines:= cl; bf.columns:= 44;
UseEntryForm( bf, notCancelled );
END HowManyRuns;
BEGIN
HowManyRuns;
FOR i:= 1 TO maxRunNo DO
SimRun;
END(*FOR*);
END DoSeveralRuns;

PROCEDURE InstallModels;
BEGIN
IF notCancelled THEN
GreenClim.UseGreenhouseClimateModel(useGreenhClimForManyYears);
IF NOT useGreenhClimForManyYears THEN
(* Weather must be first to provide correct inputs for
other submodels *)
IF useWeather THEN Weather.UseWeatherModel END;
IF usePrimProd THEN PrimProd.UsePrimaryProductionModel END;
IF useHerbivores THEN Herbivores.UseHerbivoresModel END;
IF useCarnivores THEN Carnivores.UseCarnivoresModel END;
IF useDestruents THEN Destruents.UseDestruentsModel END;
END(*IF*);
SetMonInterval( 1.0 );
SetIntegrationStep( 1.0 );
END(*IF*);
IF NOT useGreenhClimForManyYears THEN
InstallMenu(terEcoSysMenu,'Extras',enabled);
InstallCommand(terEcoSysMenu, setESKCmd,
"Set kind of agroecosystem...", ChooseEcoSysKind,
enabled, unchecked);
InstallCommand(terEcoSysMenu, setMonLevCmd,
"Set Monitoring levels...", ChooseMonitoring,
enabled, unchecked);
InstallAliasChar(terEcoSysMenu, setMonLevCmd, "L");
SetEcoSysKind(PrimProd.Maize);
END(*IF*);
DeclExperiment(DoSeveralRuns);
END InstallModels;

```

```

BEGIN
useGreenhClimForManyYears := FALSE ;
useWeather := TRUE ;
usePrimProd := TRUE ;
useHerbivores := TRUE ;
useCarnivores := TRUE ;
useDestruents := TRUE ;
ChooseModels;
AssignString( KName, bts[K].nam ); bts[K].lev := minLev;
AssignString( WName, bts[W].nam ); bts[W].lev := minLev;
AssignString( PName, bts[P].nam ); bts[P].lev := minLev;
AssignString( HName, bts[H].nam ); bts[H].lev := minLev;
AssignString( CName, bts[C].nam ); bts[C].lev := minLev;
AssignString( DName, bts[D].nam ); bts[D].lev := minLev;

IF notCancelled THEN RunSimMaster(InstallModels) END;
END TerrEcoSystem.

```

B DEFINITIONSMODULE

Primärproduzenten - Modul PrimProd

```

DEFINITION MODULE PrimProd;
(*****
Module PrimProd (Version 1.0)

```

Version written for:
MacMETH V2.5.1 (1-Pass Modula-2 implementation)
Dialog Machine V1.0 (User interface)
ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Submodel for primary production (wheat or maize)

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftliche Anwendungen) Teil A. Fischlin, SS 89". The synthesis of the whole model is made by means of the "looking outward matrix".

- List of all submodels:
- | | |
|----------------|-------------------|
| K - GreenhClim | Treibhausklima |
| W - Weather | Witterung |
| P - PrimProd | Primärproduzenten |
| H - Herbivores | Herbivore |
| C - Carnivores | Karnivore |
| D - Destruents | Destruenten |

Programming

- Design dg 20/05/89
- Implementation dg 20/05/89

Swiss Federal Institute of Technology Zurich ETHZ
Department of Environmental Sciences
Students of Systems Analysis II & Systems Ecology Group
ETH-Zentrum
CH-8092 Zurich
Switzerland

Last revision of definition: af 03/07/89

***** *)

(* Elements of Looking-Outward Matrix, to be provided for other submodels/-modules: *)

```

VAR
dmPrimProd: REAL ;

CONST
dmPrimProdLL = 0.0; dmPrimProdMin = 0.0;
dmPrimProdUL = 15000.0; dmPrimProdMax = 10000.0;
dmPrimProdDescr = "Pflanzlicher Biomasse (Trockengewicht)";
dmPrimProdIdent = "dmPrimProd";
dmPrimProdUnit = "kg(TS)/ha";

VAR
detrPrimProd: REAL ;

CONST
detrPrimProdLL = 0.0; detrPrimProdMin = 0.0;
detrPrimProdUL = 15000.0; detrPrimProdMax = 10000.0;
detrPrimProdDescr = "Pflanzlicher Detritus (Trockengewicht)";
detrPrimProdIdent = "detrPrimProd";
detrPrimProdUnit = "kg(TS)/ha/d";

VAR
consN: REAL ;

CONST
consNLL = 0.0; consNMin = 0.0;
consNUL = 5000.0; consNMax = 1000.0;
consNDescr = "Stickstoffkonsum durch Primärproduktion";
consNIdent = "consN";
consNUnit = "kg N/ha/d";

```

```

PROCEDURE UsePrimaryProductionModel ;
TYPE
EcoSysKind = (Maize, Wheat);

PROCEDURE SetPPEcoSysKind(esk: EcoSysKind);
PROCEDURE SetPrimaryProductionMonitoring(level: INTEGER);

```

END PrimProd.

Herbivore - Modul Herbivores

```

DEFINITION MODULE Herbivores;
(*****

```

Module PrimProd (Version 1.0)

Version written for:
MacMETH V2.5.1 (1-Pass Modula-2 implementation)
Dialog Machine V1.0 (User interface)
ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Submodel for herbivores

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftliche Anwendungen) Teil A. Fischlin, SS 89". The synthesis of the whole model is made by means of the "looking outward matrix".

- List of all submodels:
- | | |
|----------------|-------------------|
| K - GreenhClim | Treibhausklima |
| W - Weather | Witterung |
| P - PrimProd | Primärproduzenten |
| H - Herbivores | Herbivore |
| C - Carnivores | Karnivore |
| D - Destruents | Destruenten |

Programming
 * Design 20/05/89
 * Implementation 20/05/89
 dg

Swiss Federal Institute of Technology Zurich ETHZ
 Department of Environmental Sciences
 Students of Systems Analysis II & Systems Ecology Group
 ETH-Zentrum
 CH-8092 Zurich
 Switzerland

Last revision of definition: at 03/07/89

```
FROM PrimProd IMPORT EcosysKind;
(*
Elements of Looking-Outward Matrix,
to be provided for other submodels/-modules:
*)
```

VAR
 consHerbivores : REAL :

```
CONST
consHerbivoreall = 0.0; consHerbivoremax = 0.0;
consHerbivoreall = 500.0; consHerbivoremax = 100.0;
consHerbivoredesc = "Herbivorenkonsum (Trockensubstanz)";
consHerbivoreresident = "consHerbivores";
consHerbivoreresident = "kg(tS)/ha/d";
```

VAR
 danderbivores : REAL :

```
CONST
danderbivoreall = 0.0; danderbivoremax = 0.0;
danderbivoreall = 1500.0; danderbivoremax = 1000.0;
danderbivoredesc = "Herbivoren Biomasse (Trockensubstanz)";
danderbivoreresident = "danderbivores";
danderbivoreresident = "kg(tS)/ha/d";
```

VAR
 detHerbivores : REAL :

```
CONST
detHerbivoreall = 0.0; detHerbivoremax = 0.0;
detHerbivoreall = 100.0; detHerbivoremax = 50.0;
detHerbivoredesc = "Herbivoren Detritus (Trockensubstanz)";
detHerbivoreresident = "detHerbivores";
detHerbivoreresident = "kg(tS)/ha/d";
```

PROCEDURE UserHerbivoreModel :

PROCEDURE SetEcosysKind(pk: EcosysKind);
 PROCEDURE SetHerbivoreMonitoring(level: INTEGER);

END Herbivores;

Karnivore - Modul Carnivores

DEFINITION MODULE Carnivores :

```
Module Carnivores (Version 1.0)
Version written for:
MACHERN V2.5.1 (1-Pass Modula-2 Implementation)
Dialog Machine V1.0 (user interface)
ModelWorks V1.2 (Modelling & Simulation environment)
```

Purpose: Submodel for carnivores

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftenliche Anwendungen) Teil A, Fischlin, SS 89". The synthesis of the whole model is made by means of the "Looking outward matrix".

List of all submodels:

K - GreenhCLim	Treibhausklima
M - Weather	Witterung
F - PrimProd	Primärproduzenten
H - Herbivores	Herbivore
C - Carnivores	Karnivore
D - Destruents	Destruenten

Programming
 * Design 20/05/89
 * Implementation 20/05/89
 dg

Swiss Federal Institute of Technology Zurich ETHZ
 Department of Environmental Sciences
 Students of Systems Analysis II & Systems Ecology Group
 ETH-Zentrum
 CH-8092 Zurich
 Switzerland

Last revision of definition: or 03/07/89

FROM PrimProd IMPORT EcosysKind;

```
(*
Elements of Looking-Outward Matrix,
to be provided for other submodels/-modules:
*)
```

VAR
 consCarnivores : REAL :

```
CONST
consCarnivoreall = 0.0; consCarnivoremax = 0.0;
consCarnivoreall = 150.0; consCarnivoremax = 50.0;
consCarnivoredesc = "Karnivorenkonsum (Trockensubstanz)";
consCarnivoreresident = "consCarnivores";
consCarnivoreresident = "kg(tS)/ha/d";
```

VAR
 detCarnivores : REAL :

```
CONST
detCarnivoreall = 0.0; detCarnivoremax = 0.0;
detCarnivoreall = 50.0; detCarnivoremax = 10.0;
detCarnivoredesc = "Karnivoren Detritus (Trockensubstanz)";
detCarnivoreresident = "detCarnivores";
detCarnivoreresident = "kg(tS)/ha/d";
```

PROCEDURE UserCarnivoreModel :

PROCEDURE SetEcosysKind(pk: EcosysKind);
 PROCEDURE SetCarnivoreMonitoring(level: INTEGER);

END Carnivores;

Destruenten - Modul Destruents

DEFINITION MODULE Destruents :

```
Module PrimProd (Version 1.0)
Version written for:
MACHERN V2.5.1 (1-Pass Modula-2 Implementation)
Dialog Machine V1.0 (user interface)
ModelWorks V1.2 (Modelling & Simulation environment)
```

Purpose: Submodel for destruents

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftenliche Anwendungen) Teil A, Fischlin, SS 89". The synthesis of the whole model is made by means of the "Looking outward matrix".

List of all submodels:

K - GreenhCLim	Treibhausklima
M - Weather	Witterung
F - PrimProd	Primärproduzenten
H - Herbivores	Herbivore
C - Carnivores	Karnivore
D - Destruents	Destruenten

Programming
 * Design 20/05/89
 * Implementation 20/05/89
 dg

Swiss Federal Institute of Technology Zurich ETHZ
 Department of Environmental Sciences
 Students of Systems Analysis II & Systems Ecology Group
 ETH-Zentrum
 CH-8092 Zurich
 Switzerland

Last revision of definition: at 30/06/89

```
(*
Elements of Looking-Outward Matrix,
to be provided for other submodels/-modules:
*)
```

VAR
 NCont : REAL :

```
CONST
NContLL = 0.0; NContMin = 0.0;
NContUL = 200.0; NContMax = 80.0;
NContDesc = "N-Gehalt (mineralsiert, vertigbar)";
NContIdent = "NCont";
NContUnit = "kg/ha";
```

PROCEDURE UseDestruentsModel :

PROCEDURE SetDestruentsMonitoring(level: INTEGER);

END Destruents;

Witterung - Modul Weather

DEFINITION MODULE Weather :

```

Module PrimProd (Version 1.0)

Version written for:
MacMETH V2.5.1 (1-Pass Modula-2 implementation)
Dialog Machine V1.0 (User interface)
ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Submodel for weather

It is part of the case study model developed
in the course "Systemanalyse II (Naturwissen-
schaftliche Anwendungen) Teil A. Fischlin, SS 89".
The synthesis of the whole model is made by
means of the "looking outward matrix".

List of all submodels:
K - GreenhClim Treibhausklima
W - Weather Witterung
P - PrimProd Primärproduzenten
H - Herbivores Herbivore
C - Carnivores Karnivore
D - Destruents Destruenten

Programming
* Design dg 20/05/89
* Implementation dg 20/05/89

Swiss Federal Institute of Technology Zurich ETHZ
Department of Environmental Sciences
Students of Systems Analysis II & Systems Ecology Group
ETH-Zentrum
CH-8092 Zurich
Switzerland

Last revision of definition: af 30/06/89
    
```

Swiss Federal Institute of Technology Zurich ETHZ
 Department of Environmental Sciences
 Students of Systems Analysis II & Systems Ecology Group
 ETH-Zentrum
 CH-8092 Zurich
 Switzerland

Last revision of definition: af 30/06/89

```

(*
Elements of Looking-Outward Matrix,
to be provided for other submodels/-modules:
*)

VAR
curYear: REAL ;

CONST
curYearLL = 1800.0; curYearMin = 1950.0;
curYearUL = 2200.0; curYearMax = 2050.0;
curYearDescr = "Laufendes Jahr";
curYearIdent = "curYear";
curYearUnit = "a";

VAR
annualMeanT: REAL ;

CONST
annualMeanTLL = -10.0; annualMeanTMin = 7.5;
annualMeanTUL = +40.0; annualMeanTMax = 12.5;
annualMeanTDescr = "Globale Lufttemperatur Jahresmittel";
annualMeanTIdent = "annualMeanT";
annualMeanTUnit = "°C";

PROCEDURE UseGreenhouseClimateModel (forManyYears: BOOLEAN);
PROCEDURE SetGreenhouseClimateMonitoring(level: INTEGER);

END GreenhClim.
    
```

```

(*
Elements of Looking-Outward Matrix,
to be provided for other submodels/-modules:
*)

VAR
dTmin : REAL;

CONST
dTminLL = -50.0; dTminMin = -5.0;
dTminUL = 70.0; dTminMax = 25.0;
dTminDescr = "Tagestemperatur Minimum";
dTminIdent = "dTmin";
dTminUnit = "°C";

VAR
dTmax: REAL ;

CONST
dTmaxLL = -50.0; dTmaxMin = -5.0;
dTmaxUL = 70.0; dTmaxMax = 25.0;
dTmaxDescr = "Tagestemperatur Maximum";
dTmaxIdent = "dTmax";
dTmaxUnit = "°C";

PROCEDURE UseWeatherModel ;
PROCEDURE SetWeatherMonitoring(level: INTEGER);

END Weather.
    
```

C IMPLEMENTATIONS MODULE

Primärproduzenten - Modul PrimProd

```

IMPLEMENTATION MODULE PrimProd ;

Version written for:
MacMETH V2.5.1 (1-Pass Modula-2 implementation)
Dialog Machine V1.0 (User interface)
ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Submodel for primary production (wheat or maize)

It is part of the case study model developed
in the course "Systemanalyse II (Naturwissen-
schaftliche Anwendungen) Teil A. Fischlin, SS 89".
The synthesis of the whole model is made by
means of the "looking outward matrix"

List of all submodels:
K - GreenhClim Treibhausklima
W - Weather Witterung
P - PrimProd Primärproduzenten
H - Herbivores Herbivore
C - Carnivores Karnivore
D - Destruents Destruenten

Remarks: Systems Ecology version
    
```

Treibhausklima - Modul GreenhClim

```

DEFINITION MODULE GreenhClim ;

Module GreenhClim (Version 1.0)

Version written for:
MacMETH V2.5.1 (1-Pass Modula-2 implementation)
Dialog Machine V1.0 (User interface)
ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Submodel for the greenhouse effect

It is part of the case study model developed
in the course "Systemanalyse II (Naturwissen-
schaftliche Anwendungen) Teil A. Fischlin, SS 89".
The synthesis of the whole model is made by
means of the "looking outward matrix".

List of all submodels:
K - GreenhClim Treibhausklima
W - Weather Witterung
P - PrimProd Primärproduzenten
H - Herbivores Herbivore
C - Carnivores Karnivore
D - Destruents Destruenten

Programming
* Design dg 20/05/89
* Implementation dg 20/05/89
    
```

Implementation and Revisions:

Author	Date	Description
dg	23/05/89	Implementation of module-structure.
dg	21/06/89	Adaptation to student's LO-Matrix.
mb, or,af	04/07/89	Final implementation.

```

FROM SimBase IMPORT
Model, IntegrationMethod, DeclM, SetDefltM,
DeclSV,
RTCType, DeclP, SetP,
StashFiling, Tabulation, Graphing, DeclMV,
CurrentStep, CurrentTime,
SetSimTime, SetMonInterval, SetIntegrationStep,
TerminateConditionProcedure, InstallTerminateCondition,
StashFileName, DeclExperiment, SetMV,
GetGlobSimPars, SetGlobSimPars ;

FROM SimMaster IMPORT RunSimMaster, SimRun ;

FROM MathLib IMPORT Exp;

FROM Weather IMPORT dTmin, dTmax;

FROM Destruents IMPORT NCont;

FROM Herbivores IMPORT consHerbivores;

VAR
    
```

```

m: Model ;

Temp,
DD, dDD : REAL;
rT       : REAL;
T0, T3   : REAL;
Topt1    : REAL;
Topt2    : REAL;
DDh      : REAL;
DDf      : REAL;
stage    : REAL;
dmSh, dDmSh : REAL;
dmRt, dDmRt : REAL;
growth   : REAL;
photos   : REAL;
maint    : REAL;
rmaint,  (* ratio of maintenance *)
sla,    (* specific leaf area *)
cvf,    (* conversion factor *)
gphot,  (* maximal photosynthesis *)
fsh,    (* fraction of growth to shoot *)
ext,    (* light extinction *)
dr,     (* maximal death rate *)
Ndem,   (* N demand *)
NConsume, (* N consumption *)
consH,  (* effective consumption by herbivores *)
NConcInPl, (* N concentration in the plant *)
fN      (* fraction of consumption to demand *)
: REAL;

(* Photosynthese & Erhaltung: *)
photos:= gphot* (1.0-Exp(-ext*sla*dmSh)) *dDD;
maint := (dmSh+dmRt)* rmaint;

(* N Bedarf: *)
Ndem:= photos * NConcInPl;
NConsume:= Min(Ndem,NCont);
IF Ndem > 0.0 THEN
  fN:= NConsume / Ndem;
ELSE
  fN:=0.0
END(*IF*);

detrPrimProd:= stage*dmPrimProd* dr;
ELSE
  photos:= 0.0;
  maint := 0.0;
  detrPrimProd:= 0.0;
  fN := 0.0;
END(*IF*);

(* Eigentliche Output Variablen: *)
consN := fN* Ndem;

consH := Min( consHerbivores,
             dmPrimProd-detrPrimProd*(fN*photos-maint)*cvf );

(* Ratenvariablen: *)
growth:= (fN*photos-maint)*cvf -detrPrimProd -consH;
dDmSh := fsh*growth;
dDmRt := (1.0-fsh)*growth;

dmPrimProd := dmSh+ dmRt;

END Dynamic;

PROCEDURE Max( x1, x2 : REAL ) : REAL;
BEGIN
  IF x1 > x2 THEN RETURN x1;
  ELSE RETURN x2;
END(*IF*);
END Max;

PROCEDURE Min( x1, x2 : REAL ) : REAL;
BEGIN
  IF x1 < x2 THEN RETURN x1;
  ELSE RETURN x2;
END(*IF*);
END Min;

PROCEDURE AboutModel; BEGIN END AboutModel;
PROCEDURE Initial; BEGIN END Initial;
PROCEDURE Terminal; BEGIN END Terminal;
PROCEDURE Input; BEGIN END Input;

PROCEDURE Output;
BEGIN
  (* Temperaturabhängigkeit: *)
  Temp := (dTmin+dTmax)/2.0;
  IF (Temp <= T0) OR (Temp >= T3 ) THEN
    rT:= 0.0;
  ELSIF (Temp > T0) AND (Temp <= Topt1) THEN
    rT:= (Temp-T0)/(Topt1-T0);
  ELSIF (Temp > Topt1) AND (Temp <= Topt2) THEN
    rT:= 1.0;
  ELSE
    rT:= (T3-Temp)/(T3-Topt2);
  END(*IF*);

  dDD := rT* (Topt1-T0);

  IF dDD > 0.0 THEN
    IF (DD > DDh) THEN stage:= 1.0;
    ELSIF (DD > DDf) THEN stage:= 0.5;
    ELSE stage:= 0.0;
  END(*IF*);

  (* Photosynthese & Erhaltung: *)
  photos:= gphot* (1.0-Exp(-ext*sla*dmSh)) *dDD;
  maint := (dmSh+dmRt)* rmaint;

  (* N Bedarf: *)
  Ndem:= photos * NConcInPl;
  NConsume:= Min(Ndem,NCont);
  IF Ndem > 0.0 THEN
    fN:= NConsume / Ndem;
  ELSE
    fN:=0.0
  END(*IF*);
  ELSE
    photos:= 0.0;
    maint := 0.0;
    fN := 0.0;
  END(*IF*);

  (* Eigentliche Output Variablen: *)
  consN:= fN* Ndem;

  detrPrimProd:= stage*dmPrimProd* dr;

  dmPrimProd := dmSh+ dmRt;
END Output;

PROCEDURE Dynamic;
BEGIN
  (* Temperaturabhängigkeit: *)
  Temp := (dTmin+dTmax)/2.0;
  dDD := Max( Temp-T0, 0.0 );

  IF dDD > 0.0 THEN
    IF (DD > DDh) THEN stage:= 1.0;
    ELSIF (DD > DDf) THEN stage:= 0.5;
    ELSE stage:= 0.0;
  END(*IF*);
  ELSE
    photos:= 0.0;
    maint := 0.0;
    detrPrimProd:= 0.0;
    fN := 0.0;
  END(*IF*);

  (* State variables: *)
  DeclSV( DD, dDD, 0.0, 0.0, 1500.0,
          'Temperatursumme', 'DD', '°C*d' );
  DeclMV( DD, 0.0, 1600.0,
          'Temperatursumme', 'DD', '°C*d',
          notOnFile, writeInTable, isY );
  DeclMV( dDD, 0.0, 20.0,
          'DD Zunahme', 'dDD', '°C',
          notOnFile, notInTable, notInGraph );

  DeclSV( dmSh, dDmSh, 100.0, 0.0, 6000.0,
          'Sprossgewicht (Trockensubstanz)', 'dmSh', 'kg(TS)/ha' );
  DeclMV( dmSh, 0.0, 20000.0,
          'Sprossgewicht (Trockensubstanz)', 'dmSh', 'kg(TS)/ha',
          notOnFile, writeInTable, isY );
  DeclMV( dDmSh, 0.0, 20.0,
          'Zunahme Sprossgewicht (Trockensubstanz)',
          'dDmSh', 'kg(TS)/ha',
          notOnFile, notInTable, notInGraph );

  DeclSV( dmRt, dDmRt, 50.0, 0.0, 6000.0,
          'Wurzelgewicht (Trockensubstanz)', 'dmRt', 'kg(TS)/ha' );
  DeclMV( dmRt, 0.0, 20000.0,
          'Wurzelgewicht (Trockensubstanz)', 'dmRt', 'kg(TS)/ha',
          notOnFile, writeInTable, isY );
  DeclMV( dDmRt, 0.0, 20.0,
          'Zunahme Wurzelgewicht (Trockensubstanz)',
          'dDmRt', 'kg(TS)/ha',
          notOnFile, notInTable, notInGraph );

  (* Parameters: *)
  DeclP( T0, 6.0, 0.0, 15.0, rtc,
         'unterer Entwicklungsnullpunkt', 'T0', '°C' );
  DeclP( Topt1, 20.0, 0.0, 30.0, rtc,
         'unteres Entwicklungsoptimum', 'Topt1', '°C' );
  DeclP( Topt2, 25.0, 0.0, 30.0, rtc,
         'oberes Entwicklungsoptimum', 'Topt2', '°C' );
  DeclP( T3, 35.0, 10.0, 30.0, rtc,
         'oberer Entwicklungsnullpunkt', 'T3', '°C' );

  DeclP( DDf, 600.0, 0.0, 1500.0, rtc,
         'Blüte', 'DDf', '°C*d' );
  DeclP( DDh, 1500.0, 0.0, 2000.0, rtc,
         'Reife', 'DDh', '°C*d' );

  DeclP( ext, 0.80, 0.0, 1.0, rtc,
         'Extinktions Koeff.', 'ext', '%/100' );
  DeclP( cvf, 0.70, 0.0, 1.0, rtc,
         'Konversions Koeff.', 'cvf', '%/100' );
  DeclP( rmaint, 0.015, 0.0, 1.0, rtc,
         'Erhaltungsanteil', 'rmaint', '%/100' );
  DeclP( fsh, 0.60, 0.0, 1.0, rtc,
         'Wachst.anteil Spross', 'fsh', '%/100' );
  DeclP( sla, 0.0025, 0.0, 1.0, rtc,
         'Spez. Blattfläche', 'sla', 'ha/kg' );
  DeclP( gphot, 40.0, 0.0, 1000.0, rtc,
         'Max. Photosyntheserate', 'gphot',
         'kg(TS)/(°C*ha)' );
  DeclP( dr, 0.005, 0.0, 1.0, rtc,
         'Max. Absterberate', 'dr', 'kg(TS)/(d*ha)' );
  DeclP( NConcInPl, 0.01, 0.0, 1.0, rtc,
         'N-Gehalt in Pflanzenbiomasse (Trockensubstanz)',
         'NConcInPl', '%/100' );

  (* Monitoring variables: *)
  DeclMV( dmPrimProd, 0.0, 20000.0,
         'Primärproduzenten Biomasse (Trockensubstanz)',
         'dmPrimProd', 'kg(TS)/ha',
         notOnFile, notInTable, isY );
  DeclMV( detrPrimProd, 0.0, 100.0,
         'Primärproduzenten Detritus (Trockensubstanz)',

```



```
'detrPrimProd', 'kg(TS)/ha',
notOnFile, notInTable, isY );

DeclMV( T0, -10.0, 30.0,
'EntwicklungsNullpunkt', 'T0', '°C',
notOnFile, notInTable, notInGraph );
DeclMV( stage, 0.0, 1.2,
'Reifezeitpunkt', 'stage', '-',
notOnFile, notInTable, isY );

DeclMV( maint, 0.0, 1000.0,
'Erhaltungsrespiration', 'maint', 'kg(TS)/(d*ha)',
notOnFile, notInTable, isY );
DeclMV( photos, 0.0, 1000.0,
'Photosyntheserate', 'photos', 'kg(TS)/(d*ha)',
notOnFile, notInTable, isY );

DeclMV( NDem, 0.0, 10.0,
'N-Bedarf', 'NDem', 'kg/(d*ha)',
notOnFile, notInTable, isY );
DeclMV( NConcInPl, 0.0, 1.0,
'N-Gehalt in Pflanzenbiomasse (Trockensubstanz)',
'NConcInPl', '%/100',
notOnFile, notInTable, notInGraph );
```

END Objects;

```
PROCEDURE UsePrimaryProductionModel ;
BEGIN
```

```
DeclM (m, Euler, Initial, Input, Output, Dynamic, Terminal,
Objects,
'Untermmodell Mais', 'Maize', AboutModel);
SetSimTime( 80.0, 220.0 );
```

END UsePrimaryProductionModel;

```
PROCEDURE SetPPEcoSysKind(esk: EcoSysKind);
BEGIN
```

```
CASE esk OF
Maize:
```

```
SetP( m, T0, 6.0 );
SetP( m, Topt1, 20.0 );
SetP( m, Topt2, 25.0 );
SetP( m, T3, 35.0 );
SetP( m, DDF, 600.0 );
SetP( m, DDh, 1500.0 );
SetP( m, ext, 0.8 );
SetP( m, cvf, 0.7 );
SetP( m, rmaint, 0.015 );
SetP( m, fsh, 0.6 );
SetP( m, sla, 0.0025 );
SetP( m, gphot, 40.0 );
SetP( m, dr, 0.005 );
SetP( m, NConcInPl, 0.01 );
SetDeflTM( m, Euler,
Initial, Input, Output, Dynamic, Terminal,
'Untermmodell Mais', 'Maize', AboutModel);
```

```
| Wheat:
```

```
SetP( m, T0, 4.0 );
SetP( m, Topt1, 15.0 );
SetP( m, Topt2, 20.0 );
SetP( m, T3, 30.0 );
SetP( m, DDF, 500.0 );
SetP( m, DDh, 1000.0 );
SetP( m, ext, 0.8 );
SetP( m, cvf, 0.7 );
SetP( m, rmaint, 0.015 );
SetP( m, fsh, 0.6 );
SetP( m, sla, 0.002 );
SetP( m, gphot, 30.0 );
SetP( m, dr, 0.005 );
SetP( m, NConcInPl, 0.01 );
SetDeflTM( m, Euler,
Initial, Input, Output, Dynamic, Terminal,
'Untermmodell Weizen', 'Wheat', AboutModel);
```

END(*CASE*);
END SetPPEcoSysKind;

```
PROCEDURE SetPrimaryProductionMonitoring (level: INTEGER);
BEGIN
```

```
CASE level OF
0:
```

```
SetMV( m, Temp, -10.0, 30.0,
notOnFile, notInTable, notInGraph );
SetMV( m, DD, 0.0, 1600.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dDD, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmSh, 0.0, 25000.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dDmSh, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmRt, 0.0, 25000.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dDmRt, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmPrimProd, 0.0, 25000.0,
notOnFile, writeInTable, isY );
SetMV( m, detrPrimProd, 0.0, 100.0,
notOnFile, notInTable, notInGraph );
SetMV( m, T0, -10.0, 30.0,
notOnFile, notInTable, notInGraph );
SetMV( m, stage, 0.0, 1.2,
notOnFile, notInTable, notInGraph );
SetMV( m, maint, 0.0, 1000.0,
notOnFile, notInTable, notInGraph );
SetMV( m, photos, 0.0, 1000.0,
notOnFile, notInTable, notInGraph );
SetMV( m, NDem, 0.0, 10.0,
notOnFile, notInTable, notInGraph );
SetMV( m, NConcInPl, 0.0, 1.0,
notOnFile, notInTable, notInGraph );
```

| 1:

```
SetMV( m, Temp, -10.0, 30.0,
notOnFile, notInTable, notInGraph );
SetMV( m, DD, 0.0, 1600.0,
notOnFile, writeInTable, isY );
SetMV( m, dDD, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmSh, 0.0, 25000.0,
notOnFile, writeInTable, isY );
SetMV( m, dDmSh, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmRt, 0.0, 25000.0,
notOnFile, writeInTable, isY );
SetMV( m, dDmRt, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmPrimProd, 0.0, 25000.0,
notOnFile, notInTable, isY );
SetMV( m, detrPrimProd, 0.0, 100.0,
notOnFile, notInTable, isY );
SetMV( m, T0, -10.0, 30.0,
notOnFile, notInTable, notInGraph );
SetMV( m, stage, 0.0, 1.2,
notOnFile, notInTable, isY );
SetMV( m, maint, 0.0, 1000.0,
notOnFile, notInTable, isY );
SetMV( m, photos, 0.0, 1000.0,
notOnFile, notInTable, isY );
SetMV( m, NDem, 0.0, 10.0,
notOnFile, notInTable, isY );
SetMV( m, NConcInPl, 0.0, 1.0,
notOnFile, notInTable, notInGraph );
```

| 2:

```
SetMV( m, Temp, -10.0, 30.0,
notOnFile, notInTable, isY );
SetMV( m, DD, 0.0, 1600.0,
notOnFile, writeInTable, isY );
SetMV( m, dDD, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmSh, 0.0, 25000.0,
notOnFile, writeInTable, isY );
SetMV( m, dDmSh, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmRt, 0.0, 25000.0,
notOnFile, writeInTable, isY );
SetMV( m, dDmRt, 0.0, 20.0,
notOnFile, notInTable, notInGraph );
SetMV( m, dmPrimProd, 0.0, 25000.0,
notOnFile, notInTable, isY );
SetMV( m, detrPrimProd, 0.0, 100.0,
notOnFile, notInTable, isY );
SetMV( m, T0, -10.0, 30.0,
notOnFile, notInTable, notInGraph );
SetMV( m, stage, 0.0, 1.2,
notOnFile, notInTable, isY );
SetMV( m, maint, 0.0, 1000.0,
notOnFile, notInTable, isY );
SetMV( m, photos, 0.0, 1000.0,
notOnFile, notInTable, isY );
SetMV( m, NDem, 0.0, 10.0,
notOnFile, notInTable, isY );
SetMV( m, NConcInPl, 0.0, 1.0,
notOnFile, notInTable, notInGraph );
```

ELSE

END(*CASE*);

END SetPrimaryProductionMonitoring;

END PrimProd.

Herbivore - Modul *Herbivores*

IMPLEMENTATION MODULE Herbivores :

(*****)

Version written for:
 MacMETH V2.5.1 (1-Pass Modula-2 implementation)
 Dialog Machine V1.0 (User interface)
 ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Submodel for herbivores

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftliche Anwendungen) Teil A. Fischlin, SS 89". The synthesis of the whole model is made by means of the "looking outward matrix".

List of all submodels:
 K - GreenhClim Treibhausklima
 W - Weather Witterung
 P - PrimProd Primärproduzenten
 H - Herbivores Herbivore
 C - Carnivores Karnivore
 D - Destruents Destruenten

Remarks: Systems Ecology version

Implementation and Revisions:

Author	Date	Description
dg	23/05/89	Implementation of module-structure.
dg	21/06/89	Adaptation to student's IO-Matrix.
af	26/06/89	Final version for release
or	04/07/89	Final implementation.

(*****)

FROM SimBase IMPORT
 Model, IntegrationMethod, DeclM, SetDeflTM,
 Declsv,

```

RTCType, DeclP, SetP,
StashFiling, Tabulation, Graphing, DeclMV,
CurrentStep, CurrentTime,
SetSimTime, SetMonInterval, SetIntegrationStep,
TerminateConditionProcedure, InstallTerminateCondition,
StashFileName, DeclExperiment, SetMV ;

FROM SimMaster IMPORT RunSimMaster, SimRun ;
FROM PrimProd IMPORT dmPrimProd, EcoSysKind ;
FROM Weather IMPORT dTmin, dTmax ;
FROM Carnivores IMPORT consCarnivores ;

VAR
m: Model ;
[* dmHerbivores *]
dDmHerbivores,
cvf,
beta,
dr,
fmax, f0,
Temp, T0, Topt1, Topt2, T3,
rT, f1B: REAL;

PROCEDURE AboutModel: BEGIN END AboutModel;
PROCEDURE Initial: BEGIN END Initial;
PROCEDURE Input: BEGIN END Input;
PROCEDURE Terminal: BEGIN END Terminal;

PROCEDURE Min (x1, x2:REAL):REAL;
BEGIN
IF x1 < x2 THEN RETURN x1;
ELSE RETURN x2;
END(*IF*);
END Min;

PROCEDURE Max (x1, x2:REAL):REAL;
BEGIN
IF x1 > x2 THEN RETURN x1;
ELSE RETURN x2;
END(*IF*);
END Max;

PROCEDURE Output:
BEGIN
[* Temperaturabhängigkeit: *]
Temp := (dTmin+dTmax)/2.0;
IF (Temp <= T0) OR (Temp >= T3) THEN
rT:= 0.0;
ELSIF (Temp > T0) AND (Temp <= Topt1) THEN
rT:= (Temp-T0)/(Topt1-T0);
ELSIF (Temp > Topt1) AND (Temp <= Topt2) THEN
rT:= 1.0;
ELSE
rT:= (T3-Temp)/(T3-Topt2);
END(*IF*);

[* Konsum (Funktionale Reaktion) nach Michaelis-Menten: *]
f1B := fmax* dmPrimProd/(dmPrimProd+beta);
consHerbivores := Min( dmHerbivores * f1B*rT, dmPrimProd );

[* Sterberate: *]
detrHerbivores := dmHerbivores * dr;

END Output;

PROCEDURE Dynamic:
BEGIN
[* Konsum (Funktionale Reaktion) nach Michaelis-Menten: *]
f1B := fmax* dmPrimProd/(dmPrimProd+beta);
consHerbivores := Min( dmHerbivores * f1B*rT, dmPrimProd );

[* Sterberate: *]
detrHerbivores := dmHerbivores * dr;

dDmHerbivores:= consHerbivores*cvf - detrHerbivores
- consCarnivores;

END Dynamic;

PROCEDURE Objects:
BEGIN
[* state variables .*]

DeclSV ( dmHerbivores, dDmHerbivores, 100.0, 0.0, 10000.0,
dmHerbivoresDescr, dmHerbivoresIdent, dmHerbivoresUnit );

[* monitorable variables .*]

DeclMV (dmHerbivores, dmHerbivoresMin, dmHerbivoresMax,
dmHerbivoresDescr, dmHerbivoresIdent, dmHerbivoresUnit,
notOnFile, notInTable, isY );

DeclMV (consHerbivores, consHerbivoresMin, consHerbivoresMax,
consHerbivoresDescr, consHerbivoresIdent,
consHerbivoresUnit,
notOnFile, notInTable, isY );

DeclMV (detrHerbivores, detrHerbivoresMin, detrHerbivoresMax,
detrHerbivoresDescr, detrHerbivoresIdent,
detrHerbivoresUnit,
notOnFile, notInTable, isY );

DeclMV (dDmHerbivores, -100.0, 200.0,
'Herbivorenzuwachs', 'dDmHerbivores', 'kg(TS)/ha/d',

```

```

notOnFile, notInTable, notInGraph );

DeclMV ( rT, 0.0, 1.0,
'Temperaturabhängiger Zuwachs', 'rT', '%/100',
notOnFile, notInTable, notInGraph );

[* model parameters .*]

DeclP ( T0, 10.0, 0.0, 15.0, rtc,
'unterer Entwicklungsnulppunkt', 'T0', '°C' );
DeclP ( Topt1, 20.0, 0.0, 30.0, rtc,
'unteres Entwicklungsoptimum', 'Topt1', '°C' );
DeclP ( Topt2, 25.0, 0.0, 30.0, rtc,
'oberes Entwicklungsoptimum', 'Topt2', '°C' );
DeclP ( T3, 35.0, 10.0, 30.0, rtc,
'oberer Entwicklungsnulppunkt', 'T3', '°C' );

DeclP ( beta, 100.0, 0.0, 1000.0, rtc,
'beta', 'beta', 'kg(TS)/ha' );
DeclP ( fmax, 0.11, 0.0, 100.0, rtc,
'fmax', 'fmax', 'kg(TS)/ha' );

DeclP ( cvf, 0.7, 0.0, 2.0, rtc,
'cvf', 'cvf', 'kg/kg' );

DeclP ( dr, 0.02, 0.0, 1.0, rtc,
'Sterberate', 'dr', '1/d' );

END Objects;

PROCEDURE UseHerbivoresModel ;
BEGIN

DeclM (m, Euler, Initial, Input, Output, Dynamic, Terminal,
Objects,
'Untermmodell Maiszünsler', 'CornBorer', AboutModel);

END UseHerbivoresModel ;

PROCEDURE SetHEcoSysKind(esk: EcoSysKind);
BEGIN
CASE esk OF
Maize:
SetP( m, T0, 10.0 );
SetP( m, Topt1, 20.0 );
SetP( m, Topt2, 25.0 );
SetP( m, T3, 35.0 );
SetP( m, beta, 100.0 );
SetP( m, fmax, 0.11 );
SetP( m, cvf, 0.7 );
SetP( m, dr, 0.02);
SetDeflM( m, Euler,
Initial, Input, Output, Dynamic, Terminal,
'Untermmodell Maiszünsler', 'CornBorer',
AboutModel);
| Wheat:
SetP( m, T0, 4.0 );
SetP( m, Topt1, 18.0 );
SetP( m, Topt2, 22.0 );
SetP( m, T3, 30.0 );
SetP( m, beta, 100.0 );
SetP( m, fmax, 0.11 );
SetP( m, cvf, 0.7 );
SetP( m, dr, 0.02);
SetDeflM( m, Euler,
Initial, Input, Output, Dynamic, Terminal,
'Untermmodell Blattläuse', 'Aphids',
AboutModel);

END(*CASE*);
END SetHEcoSysKind;

PROCEDURE SetHerbivoresMonitoring (level: INTEGER);
BEGIN
CASE level OF
0: SetMV(m, consHerbivores, consHerbivoresMin, consHerbivoresMax,
notOnFile, notInTable, notInGraph );
SetMV(m, dmHerbivores, dmHerbivoresMin, dmHerbivoresMax,
notOnFile, writeInTable, isY );
SetMV(m, detrHerbivores, detrHerbivoresMin, detrHerbivoresMax,
notOnFile, notInTable, notInGraph );
SetMV(m, dDmHerbivores, -100.0, 200.0,
notOnFile, notInTable, notInGraph );
SetMV(m, rT, 0.0, 1.0,
notOnFile, notInTable, notInGraph );
| 1: SetMV(m, consHerbivores, consHerbivoresMin, consHerbivoresMax,
notOnFile, notInTable, notInGraph );
SetMV(m, dmHerbivores, dmHerbivoresMin, dmHerbivoresMax,
notOnFile, writeInTable, isY );
SetMV(m, detrHerbivores, detrHerbivoresMin, detrHerbivoresMax,
notOnFile, notInTable, notInGraph );
SetMV(m, dDmHerbivores, -100.0, 200.0,
notOnFile, notInTable, isY );
SetMV(m, rT, 0.0, 1.0,
notOnFile, notInTable, isY );
| 2: SetMV(m, consHerbivores, consHerbivoresMin, consHerbivoresMax,
notOnFile, writeInTable, isY );
SetMV(m, dmHerbivores, dmHerbivoresMin, dmHerbivoresMax,
notOnFile, writeInTable, isY );
SetMV(m, detrHerbivores, detrHerbivoresMin, detrHerbivoresMax,
notOnFile, writeInTable, isY );
SetMV(m, dDmHerbivores, -100.0, 200.0,
notOnFile, notInTable, isY );
SetMV(m, rT, 0.0, 1.0,
notOnFile, notInTable, isY );

ELSE
END(*CASE*);
END SetHerbivoresMonitoring ;

END Herbivores.

```

Karnivore - Modul Carnivores

IMPLEMENTATION MODULE Carnivores ;

```

(*****

```

```

Version written for:
MacMETH V2.5.1      (1-Pass Modula-2 implementation)
Dialog Machine V1.0 (User interface)
ModelWorks V1.2     (Modelling & Simulation environment)

```

Purpose: Submodel for carnivores

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftliche Anwendungen) Teil A. Fischlin, SS 89". The synthesis of the whole model is made by means of the "looking outward matrix".

List of all submodels:

K - GreenhClim	Treibhausklima
W - Weather	Witterung
P - PrimProd	Primärproduzenten
H - Herbivores	Herbivore
C - Carnivores	Karnivore
D - Destruents	Destruenten

Remarks: Systems Ecology version

Implementation and Revisions:

Author	Date	Description
dg	23/05/89	Implementation of module-structure.
dg	21/06/89	Adaptation to student's IO-Matrix.
or	04/07/89	Finish of implementation following the students documentation.

```

(*****

```

```

FROM SimBase IMPORT
  Model, IntegrationMethod, DeclM, SetDeflM,
  DeclSV,
  RTCType, DeclP, SetP,
  StashFiling, Tabulation, Graphing, DeclMV,
  CurrentStep, CurrentTime,
  SetSimTime, SetMonInterval, SetIntegrationStep,
  TerminateConditionProcedure, InstallTerminateCondition,
  StashFileName, DeclExperiment, SetMV ;

```

```

FROM SimMaster IMPORT RunSimMaster, SimRun ;

```

```

FROM PrimProd IMPORT EcoSysKind;

```

```

FROM Herbivores IMPORT dmHerbivores ;

```

```

FROM Weather IMPORT dTmin, dTmax ;

```

VAR

```

m: Model ;
cvf, dr,
Temp,
T0, T3,
Topt1, Topt2,
rT,
a, maxConsume,
consume,
deathr,
dmCarnivores,
dDmCarnivores      : REAL;

```

```

PROCEDURE AboutModel; BEGIN END AboutModel;
PROCEDURE Input;      BEGIN END Input;
PROCEDURE Initial;    BEGIN END Initial;
PROCEDURE Terminal;   BEGIN END Terminal;

```

```

PROCEDURE Min (x1, x2:REAL):REAL;
BEGIN
  IF x1 < x2 THEN RETURN x1;
  ELSE RETURN x2;
END(*IF*);
END Min;

```

```

PROCEDURE Output;
BEGIN

```

```

(* Temperaturabhängigkeit: *)
Temp := (dTmin+dTmax)/2.0;
IF (Temp <= T0) OR (Temp >= T3) THEN
  rT:= 0.0;
ELSIF (Temp > T0) AND (Temp <= Topt1) THEN
  rT:= (Temp-T0)/(Topt1-T0);
ELSIF (Temp > Topt1) AND (Temp <= Topt2) THEN
  rT:= 1.0;
ELSE
  rT:= (T3-Temp)/(T3-Topt2);
END(*IF*);

```

```

(* Funktionale Reaktion: *)
consume := Min( dmCarnivores*dmHerbivores*a,
               dmCarnivores*maxConsume );

```

```

consCarnivores:= Min( rT*consume, dmHerbivores );

```

```

(* Sterberate: *)
deathr := dr*dmCarnivores;
detrCarnivores:= deathr;

```

END Output;

```

PROCEDURE Dynamic;
BEGIN

```

```

(* Funktionale Reaktion: *)
consume := Min( dmCarnivores*dmHerbivores*a,
               dmCarnivores*maxConsume );
consCarnivores:= Min( rT*consume, dmHerbivores );

```

```

(* Sterberate: *)
deathr := dr*dmCarnivores;
detrCarnivores:= deathr;

```

```

(* Karnivorenzuwachsrate: *)
dDmCarnivores:= cvf* consCarnivores - deathr;

```

END Dynamic;

```

PROCEDURE Objects;
BEGIN

```

(*. state variables .*)

```

DeclSV ( dmCarnivores , dDmCarnivores , 10.0 , 0.0 , 1000.0 ,
         'Karnivoren Biomasse (Trockensubstanz)',
         'dmCarnivores', 'kg(TS)/ha' );

```

(*. monitorable variables .*)

```

DeclMV (consCarnivores, consCarnivoresMin, consCarnivoresMax,
        consCarnivoresDescr, consCarnivoresIdent,
        consCarnivoresUnit,
        notOnFile, notInTable, isY) ;

```

```

DeclMV (detrCarnivores, detrCarnivoresMin, detrCarnivoresMax,
        detrCarnivoresDescr, detrCarnivoresIdent,
        detrCarnivoresUnit,
        notOnFile, notInTable, isY) ;

```

```

DeclMV ( dmCarnivores , 0.0 , 250.0 ,
         'Karnivoren Biomasse (Trockensubstanz)',
         'dmCarnivores', 'kg(TS)/ha',
         notOnFile, notInTable, isY );

```

```

DeclMV ( dDmCarnivores , 0.0 , 5.0 ,
         'KarnivorenZuwachsrate', 'dDmCarnivores', 'kg(TS)/ha/d',
         notOnFile, notInTable, isY );

```

```

DeclMV ( rT , 0.0 , 1.0 ,
         'Temperaturabhängiger Zuwachs', 'rT', '%/100',
         notOnFile, notInTable, isY );

```

```

DeclMV ( consume , 0.0 , 0.5 ,
         'Konsum', 'consume', 'kg(TS)/kg(TS)/d',
         notOnFile, notInTable, notInGraph );

```

```

DeclMV ( deathr , 0.0 , 1.0 ,
         'Sterberate', 'deathr', 'kg(TS)/d',
         notOnFile, notInTable, notInGraph );

```

```

DeclMV ( Temp , 0.0 , 30.0 ,
         'Temperatur', 'Temp', '°C',
         notOnFile, notInTable, notInGraph );

```

(*. model parameters .*)

```

DeclP ( T0 , 12.0 , 0.0 , 15.0 , rtc,
        'uUnterer Entwicklungsnulppunkt', 'T0', '°C' );

```

```

DeclP ( Topt1 , 25.0 , 0.0 , 30.0 , rtc,
        'unteres Entwicklungsoptimum', 'Topt1', '°C' );

```

```

DeclP ( Topt2 , 30.0 , 0.0 , 30.0 , rtc,
        'oberes Entwicklungsoptimum', 'Topt2', '°C' );

```

```

DeclP ( T3 , 35.0 , 10.0 , 30.0 , rtc,
        'oberer Entwicklungsnulppunkt', 'T3', '°C' );

```

```

DeclP ( a , 0.005 , 0.0 , 1.0 , rtc,
        'Funkt.Reakt.Steigung', 'a', 'kg(TS)/kg(TS)/d' );

```

```

DeclP ( maxConsume , 0.2 , 0.0 , 1.0 , rtc,
        'Maximalkonsum pro Herbivor', 'maxConsume', 'kg(TS)/d' );

```

```

DeclP ( cvf , 0.70 , 0.0 , 1.0 , rtc,
        'Wachstumsausnützung', 'cvf', 'kg(TS)/kg(TS)' );

```

```

DeclP ( dr , 0.02 , 0.0 , 1.0 , rtc,
        'Sterberate', 'dr', '1/d' );

```

END Objects;

```

PROCEDURE UseCarnivoresModel ;
BEGIN

```

```

DeclM (m, Euler, Initial, Input, Output, Dynamic, Terminal,
       Objects,
       'Untermmodell Trichogramma', 'Trichogramma', AboutModel);

```

END UseCarnivoresModel ;

```

PROCEDURE SetEcoSysKind(esk: EcoSysKind);

```

BEGIN

CASE esk OF

Maize:

```

SetP( m, T0 , 12.0 );
SetP( m, Topt1, 25.0 );
SetP( m, Topt2, 30.0 );
SetP( m, T3 , 35.0 );
SetP( m, a , 0.005 );
SetP( m, maxConsume, 0.2 );
SetP( m, cvf , 0.7 );

```

```

SetP(m, dr, 0.02);
SetDeflM(m, Euler,
Initial, Input, Output, Dynamic, Terminal,
'Untermodell Trichogramma', 'Trichogramma',
AboutModel);
| Wheat:
SetP(m, T0, 8.0);
SetP(m, Topt1, 22.0);
SetP(m, Topt2, 28.0);
SetP(m, T3, 35.0);
SetP(m, a, 0.005);
SetP(m, maxConsums, 0.2);
SetP(m, cvf, 0.7);
SetP(m, dr, 0.02);
SetDeflM(m, Euler,
Initial, Input, Output, Dynamic, Terminal,
'Untermodell Coccinellidenlarven',
'Coccinellids', AboutModel);
END(*CASE*);
END SetEcoSysKind;

PROCEDURE SetCarnivoresMonitoring (level: INTEGER);
BEGIN
CASE level OF
0: SetMV(m, consCarnivores, consCarnivoresMin,
consCarnivoresMax,
notOnFile, notInTable, notInGraph);
SetMV(m, detrCarnivores, detrCarnivoresMin,
detrCarnivoresMax,
notOnFile, notInTable, notInGraph);
SetMV(m, dmCarnivores, 0.0, 250.0,
notOnFile, writeInTable, isY);
SetMV(m, dDmCarnivores, 0.0, 5.0,
notOnFile, notInTable, notInGraph);
SetMV(m, rT, 0.0, 1.0,
notOnFile, notInTable, notInGraph);
SetMV(m, consume, 0.0, 0.5,
notOnFile, notInTable, notInGraph);
SetMV(m, deathr, 0.0, 1.0,
notOnFile, notInTable, notInGraph);
SetMV(m, Temp, 0.0, 30.0,
notOnFile, notInTable, notInGraph);
| 1: SetMV(m, consCarnivores, consCarnivoresMin,
consCarnivoresMax,
notOnFile, notInTable, notInGraph);
SetMV(m, detrCarnivores, detrCarnivoresMin,
detrCarnivoresMax,
notOnFile, notInTable, notInGraph);
SetMV(m, dmCarnivores, 0.0, 250.0,
notOnFile, notInTable, isY);
SetMV(m, dDmCarnivores, 0.0, 5.0,
notOnFile, notInTable, notInGraph);
SetMV(m, rT, 0.0, 1.0,
notOnFile, notInTable, notInGraph);
SetMV(m, consume, 0.0, 0.5,
notOnFile, writeInTable, isY);
SetMV(m, deathr, 0.0, 1.0,
notOnFile, notInTable, notInGraph);
SetMV(m, Temp, 0.0, 30.0,
notOnFile, notInTable, notInGraph);
| 2: SetMV(m, consCarnivores, consCarnivoresMin,
consCarnivoresMax,
notOnFile, notInTable, isY);
SetMV(m, detrCarnivores, detrCarnivoresMin,
detrCarnivoresMax,
notOnFile, notInTable, isY);
SetMV(m, dmCarnivores, 0.0, 250.0,
notOnFile, notInTable, isY);
SetMV(m, dDmCarnivores, 0.0, 5.0,
notOnFile, notInTable, isY);
SetMV(m, rT, 0.0, 1.0,
notOnFile, notInTable, isY);
SetMV(m, consume, 0.0, 0.5,
notOnFile, notInTable, notInGraph);
SetMV(m, deathr, 0.0, 1.0,
notOnFile, notInTable, notInGraph);
SetMV(m, Temp, 0.0, 30.0,
notOnFile, notInTable, notInGraph);
ELSE
END(*CASE*);
END SetCarnivoresMonitoring;
END Carnivores.

```

Remarks: Systems Ecology version

Implementation and Revisions:

Author	Date	Description
dg	23/05/89	Implementation of module-structure.
dg	21/06/89	Adaptation to student's LO-Matrix.
mb,or	04/07/89	Final implementation

```

*****
FROM SimBase IMPORT
Model, IntegrationMethod, DeclM,
DeclSV, SetMV,
RTCType, DeclP,
StashFiling, Tabulation, Graphing, DeclMV,
CurrentStep, CurrentTime,
SetSimTime, SetMonInterval, SetIntegrationStep,
TerminateConditionProcedure, InstallTerminateCondition,
StashFileName, DeclExperiment;

FROM SimMaster IMPORT RunSimMaster, SimRun;

FROM MathLib IMPORT Exp;

FROM Weather IMPORT dTmin, dTmax;

FROM PrimProd IMPORT detrPrimProd, consN;

FROM Herbivores IMPORT detrHerbivores;

FROM Carnivores IMPORT detrCarnivores;

VAR
m: Model;

dmDestr, dDmDestr,
N, dN,
Tmean,
dr, (* death rate *)
float, (* fraction lost *)
CConcPp, (* carbon concentration in plants *)
CConcHe, (* carbon concentration in herbivores *)
CConcCa, (* carbon concentration in carnivores *)
CPp, (* carbon from plants *)
CHe, (* carbon from herbivores *)
CCa, (* carbon from carnivores *)
CDe, (* carbon from destruents *)
CSuppl, (* carbon in the soil from Pp, He, Ca, De *)
NConcPp, (* nitrogen concentration in plants *)
NConcHe, (* nitrogen concentration in herbivores *)
NConcCa, (* nitrogen concentration in carnivores *)
NPP, (* nitrogen from plants *)
NHHe, (* nitrogen from herbivores *)
NCA, (* nitrogen from carnivores *)
NDe, (* nitrogen from destruents *)
NAir, (* nitrogen from air (rain) *)
NSuppl, (* nitrogen in the soil from Pp, He, Ca, De *)
maint, (* maintenance requirement *)
gr, (* total growth rate *)
grT, (* growth rate dependent of temperature *)
grTmax, (* maximal growth rate dependent of temperature *)
grEff, (* growth efficiency *)
al, a2, (* model parameters for the growth rate(C) eq. *)
bl, b2, b3, (* model parameters for the growth rate(T) eq. *)
minr (* mineralisation rate *)

:REAL;

```

```

PROCEDURE AboutModel; BEGIN END AboutModel;
PROCEDURE Initial;
BEGIN
grTmax := (bl*4.0*b3 - b2*b2)/(4.0*b3);
END Initial;
PROCEDURE Terminal; BEGIN END Terminal;

```

```

PROCEDURE Rmax( x1, x2 : REAL ): REAL;
BEGIN
IF x1 > x2 THEN RETURN x1;
ELSE RETURN x2;
END(*IF*);
END Rmax;

```

```

PROCEDURE Input;
BEGIN
END Input;

PROCEDURE Output;
BEGIN
(* Mineralisierter N: *)
NCont := N *minr;
END Output;

```

```

PROCEDURE Dynamic;
BEGIN
(* Temperatur: *)
Tmean := (dTmin+dTmax)/2.0;
(* Umsetzrate Temperatur abhängig (relativ): *)
grT := Rmax( (bl + b2*Tmean
+ b3*Tmean*Tmean) / grTmax, 0.0 );
(* Anfall an C aus neuem Detritus: *)
CPp := detrPrimProd * CConcPp;
CHe := detrHerbivores * CConcHe;

```

Destruenten - Modul Destruents

IMPLEMENTATION MODULE Destruents;

Version written for:
MacMETH V2.5.1 (1-Pass Modula-2 implementation)
Dialog Machine V1.0 (User interface)
ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Submodel for destruents

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftliche Anwendungen) Teil A. Fischlin, SS 89". The synthesis of the whole model is made by means of the "looking outward matrix".

- List of all submodels:
- K - GreenhClim Treibhausklima
 - W - Weather Witterung
 - P - PrimProd Primärproduzenten
 - H - Herbivores Herbivore
 - C - Carnivores Karnivore
 - D - Destruents Destruenten

```

CCA :- det:Carnlivores * CConcCA:
CDE :- dmdestr * dr*grT *(1.0-float):
CSuppl := CP * Cde +cca *Cde:

(* relative Wachstumsrate: *)
gr = grT * greff:
(* Gesamtwachstumsrate C (absolut): *)
dmdestr:= gr * CSuppl -Cde:

(* neu mineralisierter N verfügbar für Pri-Prod: *)
Ncont := N *minr:

(* Anfall an N aus neuem Detritus: *)
Np := det:PriProd * NConcPp:
Nde := det:Herblivores * NConcH:
Nca := det:Carnlivores * NConcCa:
Nde := N * grT *(1.0-float):
NSuppl:= Np * Nde + Nca * Nde *NAlr:
(* Gesamtwachstumsrate N (absolut): *)
dN := gr * Nsuppl -Nde -conah:

END Dynamic:

PROCEDURE Objects:
BEGIN
(* state variables .*)
DeclSV ( dmdestr , dmdestr , 1.20E5 , 0.0 , 2.0E5 ,
'Destruenten Biomasse (Trockensubstanz)',
'dmdestr', 'Kg(TS)/ha' );
DeclSV ( N , dN , 3000.0 , 0.0 , 1.0E4 ,
'N-Gehalt (gebunden)', 'N', 'Kg/ha' );

(* monitorable variables .*)
DeclMW (Ncont, NcontMin, NcontMax,
NcontDescr, NcontIdent, NcontUnit,
notOnFile, notInTable, ISY );
DeclMW ( dmdestr , 0.0 , 150000.0 ,
'Destruenten Biomasse (Trockensubstanz)',
'dmdestr', 'Kg(TS)/ha',
notOnFile, notInTable, ISY );
DeclMW ( N , 0.0 , 5000.0 ,
'N-Gehalt (gebunden)', 'N', 'Kg/ha',
notOnFile, notInTable, ISY );
DeclMW ( gr , 0.0 , 1.0 ,
'relative Gesamtwachstumsrate', 'gr', '%/100/d',
notOnFile, notInTable, ISY );
DeclMW ( grT , 0.0 , 1.0 ,
'r abhängige Wachstumsrate', 'grT', '%/100/d',
notOnFile, notInTable, ISY );

(* model parameters .*)
DeclP ( a1 , 0.2 , 0.0 , 1.0 , etc,
'Max (Michaëlis Menten)', 'a1', '' );
DeclP ( a2 , 0.03 , 0.0 , 100.0 , etc,
'K1/2 (Michaëlis Menten)', 'a2', '' );
DeclP ( b1 , -1.16 , -1000.0 , 1000.0 , etc,
'Wachstumsparameter', 'b1', '' );
DeclP ( b2 , 0.1907 , 0.0 , 100.0 , etc,
'Wachstumsparameter', 'b2', '' );
DeclP ( b3 , -0.0026 , 0.0 , 100.0 , etc,
'Wachstumsparameter', 'b3', '' );
DeclP ( maint , 0.001 , 0.0 , 100.0 , etc,
'Erhaltungsbedarf an C per Destruentenbiomasse',
'maint', 'Kg/Kg(TS)' );
DeclP ( dr , 0.002 , 0.0 , 1.0 , etc,
'Absterberate', 'dr', '%/100/d' );

DeclP ( CConcP , 0.98 , 0.0 , 100.0 , etc,
'C-Gehalt in Pflanzenbiomasse', 'CConcP', '%/100' );
DeclP ( CConcH , 0.89 , 0.0 , 100.0 , etc,
'C-Gehalt in Herbivorenbiomasse', 'CConcH', '%/100' );
DeclP ( CConcCa , 0.89 , 0.0 , 100.0 , etc,
'C-Gehalt in Karnivorenbiomasse', 'CConcCa', '%/100' );
DeclP ( NConcP , 0.01 , 0.0 , 100.0 , etc,
'N-Gehalt in Pflanzenbiomasse', 'NConcP', '%/100' );
DeclP ( NConcH , 0.10 , 0.0 , 100.0 , etc,
'N-Gehalt in Herbivorenbiomasse', 'NConcH', '%/100' );
DeclP ( NConcCa , 0.10 , 0.0 , 100.0 , etc,
'N-Gehalt in Karnivorenbiomasse', 'NConcCa', '%/100' );
DeclP ( NAlr , 0.10 , 0.0 , 1.0 , etc,
'N-Fixierung aus Luft', 'NAlr', 'Kg/ha/d' );
DeclP ( greff , 0.1 , 0.0 , 1.0 , etc,
'Wachstumseffizienz', 'greff', '%/100' );
DeclP ( minr , 0.003 , 0.0 , 1.0 , etc,
'N-Mineralisationsrate', 'minr', '%/100' );
DeclP ( float , 0.3 , 0.0 , 1.0 , etc,
'Destr.-Verlust', 'float', '%/100' );

END Objects:

```

```

END Objects:
PROCEDURE UsedDestruentsModel :
BEGIN
DeclM (m, Euler, Initial, Input, Output, Dynamic, Terminal,
'Untermode1 Destruenten', 'Destruents', AboutModel);
END UsedDestruentsModel :

```

```

PROCEDURE SetDestruentMonitoring (level: INTEGER):
CASE level OF
0:
SetM ( m, Ncont, NcontMin, NcontMax,
notOnFile, notInTable, notInGraph);
SetM ( m, dmdestr , 0.0 , 150000.0 ,
notOnFile, notInTable, ISY );
SetM ( m, N , 0.0 , 5000.0 ,
notOnFile, notInTable, notInGraph);
SetM ( m, gr , 0.0 , 1.0 ,
notOnFile, notInTable, notInGraph);
SetM ( m, grT , 0.0 , 1.0 ,
notOnFile, notInTable, notInGraph);
1:
SetM ( m, Ncont, NcontMin, NcontMax,
notOnFile, notInTable, ISY );
SetM ( m, dmdestr , 0.0 , 150000.0 ,
notOnFile, notInTable, ISY );
SetM ( m, N , 0.0 , 5000.0 ,
notOnFile, notInTable, ISY );
SetM ( m, gr , 0.0 , 1.0 ,
notOnFile, notInTable, ISY );
SetM ( m, grT , 0.0 , 1.0 ,
notOnFile, notInTable, ISY );
2:
SetM ( m, Ncont, NcontMin, NcontMax,
notOnFile, notInTable, ISY );
SetM ( m, dmdestr , 0.0 , 150000.0 ,
notOnFile, notInTable, ISY );
SetM ( m, N , 0.0 , 5000.0 ,
notOnFile, notInTable, ISY );
SetM ( m, gr , 0.0 , 1.0 ,
notOnFile, notInTable, ISY );
SetM ( m, grT , 0.0 , 1.0 ,
notOnFile, notInTable, ISY );
END SetDestruentMonitoring;
END Destruents.

```

END Destruents.

Witterung - Modal Weather

```

IMPLEMENTATION MODULE weather :
(* ..*)

```

Version written for: 1-Page Modula-2 Implementation
Machern V2.5.1
Dialog Machine V1.0 (user interface)
ModelWorks V1.2 (Modelling & Simulation environment)

Purpose: Submodel for weather

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftenliche Anwendungsfall) Teil A, Fischlin, SS 89". The synthesis of the whole model is made by means of the "looking outward matrix".

List of all submodels:

K - GreenKlim	Treibhausklima
W - Weather	Witterung
F - FirnProd	Firnproduzenten
H - Herblivores	Herbivore
C - Carnlivores	Karnivore
D - Destruents	Destruenten

Remarks: Systems Ecology version

Implementation and Revisions:

Author	Date	Description
dg	23/05/89	Implementation of module-structure.
dg	21/06/89	Adaptation to student's IO-Matrix.
at	27/06/89	Final version for release
at	29/06/89	First implementation of SE model
at	03/07/89	Final implementation of SE model

```

FROM Simbase IMPORT
Model, IntegrationMethod, DeclM,
DeclSV,
RTCTYP, DeclP,
Staehfling, Tabulation, Graphing, DeclMW,
CurrentStep, CurrentTime,
SetStartTime, setMonInterval, setIntegrationStep,
TerminateConditionProcedure, InstallTerminalCondition,
StaehflName, DeclExperiment,
SetM;

```

```

FROM SimMaster IMPORT RunSimMaster, SimRun :
FROM DMStrings IMPORT Concat;
FROM DMSWindow IMPORT
MouseModifiers, ClickKind, DragProc, PointClicked, RectClicked,
PointDoubleClicked, RectDoubleClicked, GetLastMouseClick,
GetCursorPosition, DoButtonReleased, Drag, SetScrollStep,
SetContentSize, GetScrollBoxPos, GetScrollBoxChange,
WindowContent, MoveOriginTo, AutoScrollProc, setScrollProc,
Hue, Color, Patline, GrayContent, Pattern, pat, black, white,
red, green, blue, cyan, magenta, yellow, setMode,
setBackgroundColor, setBackgroundColor, setColor, setPattern,
GetPattern, IdentifPos, IdentifPoint, MaxCol, MaxLin,
CellHeight, BackgroundWidth, BackgroundHeight, setBOWAction,

```

```

EraseContent, RedrawContent, SetClipping, RemoveClipping,
WindowFont, SetWindowFont, FontStyles, FontStyle, SetPos,
GetPos, ShowCaret, Invert, Write, WriteLn, WriteString,
WriteCard, WriteInt, WriteReal, WriteRealSci, SetPen, GetPen,
Dot, LineTo, Circle, Area, CopyArea, MapArea,
DisplayPredefinedPicture, StartPolygon, CloseAndFillPolygon,
QDVHSelect, QDPoint, QDRect, XYToQDPoint, RectAreaToQDRect,
SelectRestoreCopy, Turn, TurnTo, (* Move, *) ScaleUC,
ConvertPointToUC, ConvertUCToPoint, UCFrame, EraseUCFrame,
EraseUCFrameContent, SetUCPen, GetUCPen, UCdot, UCLineTo,
PlotSym;
FROM DMSSystem IMPORT TitleBarHeight;
FROM DMWindows IMPORT nonexistent, Window, WindowKind,
ScrollBars, CloseAttr, ZoomAttr, WFFixPoint, RectArea,
WindowFrame, RestoreProc, background, WindowsDone,
OuterWindowFrame, CreateWindow, UsePredefinedWindow,
RedefineWindow, RedrawTitle, DummyRestoreProc, AutoRestoreProc,
SetRestoreProc, UpdateWindow, UpdateAllWindows,
RedrawBackground, GetWindowFrame, PutOnTop, FrontWindow,
WindowExists, RemoveWindow, RemoveAllWindows;
FROM MathLib IMPORT Sqrt;

FROM RandGen IMPORT U, ResetSeeds;
FROM RandNormal IMPORT InstallU, SetPars, N;
FROM JulianDays IMPORT Jan, Feb, Mar, Apr, Mai, Jun, Jul, Aug,
Sep, Oct, Nov, Dec, TimeToDate;

FROM GreenhClim IMPORT annualMeanT, curYear;

PROCEDURE TimeToMonth(t: REAL): INTEGER;
VAR d, mo, yr, wd: INTEGER;
BEGIN
  TimeToDate(1, Jan, TRUNC(curYear), t, d, mo, yr, wd);
  RETURN mo;
END TimeToMonth;

PROCEDURE TimeToDay(t: REAL): INTEGER;
VAR d, mo, yr, wd: INTEGER;
BEGIN
  TimeToDate(1, Jan, TRUNC(curYear), t, d, mo, yr, wd);
  RETURN d;
END TimeToDay;

CONST
  tempUnit = dTminUnit;

VAR
  m: Model;
  gaTmean, (* global annual Zürich temperature mean *)
  aTmean, (* Swiss (SMA Zürich II) annual temperature mean *)
  : REAL;

  month: REAL; (* current month, only used for monitoring *)
  mTmean, (* Swiss monthly temperature current means *)
  mTmeanStar, (* Swiss monthly temperature means
over several years *)
  : ARRAY [Jan..Dec] OF REAL;
  mTmeanAct, (* Swiss monthly temperature current mean,
only used for monitoring *)
  mTstdevStar, (* Swiss monthly temperature standard deviation
over several years *)

  day: REAL; (* current day, only used for monitoring *)
  dTmean, (* Swiss daily temperature current mean *)
  dTmeanNew,
  dTstdevStar, (* Swiss daily temperature standard deviation
over several years *)
  dTtempl, (* Swiss daily temperature current amplitude *)
  dTtemplStar, (* Swiss daily temperature amplitude over
several years *)
  dTtemplstdevStar, (* Swiss daily temperature amplitude standard
deviation over several years *)
  eTstdevStar, (* Swiss daily extrema temperature standard
deviation over several years *)
  mem (* Influence of last day's temperature onto
current *)
  : REAL;
  mAbr: ARRAY [Jan..Dec] OF ARRAY [0..2] OF CHAR;
  nrIceDays, (* number of ice days where min and max < 0°C *)
  nrFrostDays, (* number of frost days where min < 0°C *)
  nrSummerDays, (* number of summer days where max >= 25°C *)
  nrHotDays, (* number of hot days where max >= hotDayThresh
(-30°C) *)
  FreezePoint, (* Threshold temperature at which water freezes *)
  hotDayThresh, (* Threshold temperature above which a day
is considered hot *)

  resSeeds (* Flag to control resetting of random
number
generator seeds in Initial *)
  : REAL;

TYPE
  StatRec = RECORD Sx, Sxx: REAL; n: INTEGER END;
VAR
  mT: ARRAY [Jan..Dec] OF StatRec;
  dT, dTn, dTx: StatRec;
  statWindow: Window; showStat: BOOLEAN;

PROCEDURE AboutModel; BEGIN END AboutModel;
PROCEDURE Input; BEGIN END Input;

PROCEDURE Terminal;
CONST marg = 2; rfw = 8; deccdig = 1; rundef = -999.99;
VAR m, maxm: [Jan..Dec]; wf: WindowFrame;
PROCEDURE Mean(Sx: REAL; n: INTEGER): REAL;
BEGIN
  IF n>0 THEN
    RETURN Sx/FLOAT(n)
  ELSE
    RETURN rundef
  END(*IF*);
END Mean;
PROCEDURE Stddev(Sx, Sxx: REAL; n: INTEGER): REAL;
BEGIN
  IF n>0 THEN
    RETURN Sqrt((Sxx-Sx*Sx)/FLOAT(n))/FLOAT(n-1)
  ELSE
    RETURN rundef
  END(*IF*);
END Stddev;

PROCEDURE Initial;
VAR m: [Jan..Dec];
BEGIN
  IF resSeeds<0.0 THEN ResetSeeds END;
  SetPars(0.0, mTstdevStar);
  gaTmean := annualMeanT;
  FOR m:= Jan TO Dec DO
    mTmean[m] := gaTmean - 15.0*(globales, langjähriges Mittel)
    + mTmeanStar[m] (* + N() *);
  WITH mT[m] DO
    Sx:= 0.0; Sxx:= 0.0; n:= 0;
  END(*WITH*);
END(*FOR*);
nrIceDays := 0.0; nrFrostDays := 0.0;
nrSummerDays := 0.0; nrHotDays := 0.0;
WITH dT DO
  Sx:= 0.0; Sxx:= 0.0; n:= 0;
END(*WITH*);
WITH dTn DO
  Sx:= 0.0; Sxx:= 0.0; n:= 0;
END(*WITH*);
WITH dTx DO
  Sx:= 0.0; Sxx:= 0.0; n:= 0;
END(*WITH*);
END Initial;

PROCEDURE Dynamic;
BEGIN
  SetPars(0.0, mTstdevStar);
  mTmeanAct := mTmean[TRUNC(month)] + N();
  SetPars(0.0, dTstdevStar*(1.0+mem));
  dTmeanNew := mem*dTmean + (1.0-mem)*mTmeanAct + N();
END Dynamic;

PROCEDURE Output;
BEGIN
  month:= FLOAT(TimeToMonth(CurrentTime));
  day:= FLOAT(TimeToDay(CurrentTime));
  SetPars(dTtemplStar, dTtemplstdevStar); dTtempl := N();
  dTmin:= dTmean - 0.5*dTtempl;
  dTmax:= dTmean + 0.5*dTtempl;
  IF (dTmin < FreezePoint) AND (dTmax < FreezePoint) THEN
    nrIceDays := nrIceDays + 1.0;
  END(*IF*);
  IF (dTmin < FreezePoint) THEN
    nrFrostDays := nrFrostDays + 1.0;
  END(*IF*);
  IF (dTmax >= 25.0) THEN
    nrSummerDays := nrSummerDays + 1.0;
  END(*IF*);
END Output;

ELSE
  RETURN rundef
END(*IF*);
END Terminal;

PROCEDURE Initial;
VAR m: [Jan..Dec];
BEGIN
  IF resSeeds<0.0 THEN ResetSeeds END;
  SetPars(0.0, mTstdevStar);
  gaTmean := annualMeanT;
  FOR m:= Jan TO Dec DO
    mTmean[m] := gaTmean - 15.0*(globales, langjähriges Mittel)
    + mTmeanStar[m] (* + N() *);
  WITH mT[m] DO
    Sx:= 0.0; Sxx:= 0.0; n:= 0;
  END(*WITH*);
END(*FOR*);
nrIceDays := 0.0; nrFrostDays := 0.0;
nrSummerDays := 0.0; nrHotDays := 0.0;
WITH dT DO
  Sx:= 0.0; Sxx:= 0.0; n:= 0;
END(*WITH*);
WITH dTn DO
  Sx:= 0.0; Sxx:= 0.0; n:= 0;
END(*WITH*);
WITH dTx DO
  Sx:= 0.0; Sxx:= 0.0; n:= 0;
END(*WITH*);
END Initial;

PROCEDURE Dynamic;
BEGIN
  SetPars(0.0, mTstdevStar);
  mTmeanAct := mTmean[TRUNC(month)] + N();
  SetPars(0.0, dTstdevStar*(1.0+mem));
  dTmeanNew := mem*dTmean + (1.0-mem)*mTmeanAct + N();
END Dynamic;

PROCEDURE Output;
BEGIN
  month:= FLOAT(TimeToMonth(CurrentTime));
  day:= FLOAT(TimeToDay(CurrentTime));
  SetPars(dTtemplStar, dTtemplstdevStar); dTtempl := N();
  dTmin:= dTmean - 0.5*dTtempl;
  dTmax:= dTmean + 0.5*dTtempl;
  IF (dTmin < FreezePoint) AND (dTmax < FreezePoint) THEN
    nrIceDays := nrIceDays + 1.0;
  END(*IF*);
  IF (dTmin < FreezePoint) THEN
    nrFrostDays := nrFrostDays + 1.0;
  END(*IF*);
  IF (dTmax >= 25.0) THEN
    nrSummerDays := nrSummerDays + 1.0;
  END(*IF*);
END Output;

```

```

IF (dMax >= nrHotDaysThresh ('should be 30')) THEN
nrHotDays := nrHotDays + 1.0
END(*IF*);
WITH w[TRUNC(month)] DO
Sx:= Sx+dTmean; Sxx:= Sxx+dTmean*dTmean; INC(n);
END(*WITH*);
WITH dt DO
Sx:= Sx+dTmean; Sxx:= Sxx+dTmean*dTmean; INC(n);
END(*WITH*);
WITH dtn DO
Sx:= Sx+dTmin; Sxx:= Sxx+dTmin*dTmin; INC(n);
END(*WITH*);
WITH dtx DO
Sx:= Sx+dTmax; Sxx:= Sxx+dTmax*dTmax; INC(n);
END(*WITH*);
END Output;

PROCEDURE UseWeatherModel :
BEGIN
DECLM (m, discretetime, Initial, Input, Output, Dynamic,
Terminal, Objects,
'Utermmodell Wetter', 'Weather', 'AboutModel');
SetStartTime(0.0,365.0);
END UseWeatherModel ;

PROCEDURE SetWeatherMonitoring(Level: INTEGER):
BEGIN
CASE Level OF
0: SetWV(m, month, 0.0, 12.0,
notOnFile, notInTable, notInGraph );
SetWV(m, day, 1.0, 31.0,
notOnFile, notInTable, notInGraph );

SetWV(m, hotDayThresh, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
SetWV(m, mTmeanAct, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
SetWV(m, dTmean, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
SetWV(m, dTmin, -20.0, 40.0,
notOnFile, writeInTable, isy );
SetWV(m, dTmax, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
SetWV(m, nrIceDays, 0.0, 365.0,
notOnFile, notInTable, notInGraph );
SetWV(m, nrFrostDays, 0.0, 365.0,
notOnFile, notInTable, notInGraph );
SetWV(m, nrSummerDays, 0.0, 365.0,
notOnFile, notInTable, notInGraph );
SetWV(m, nrHotDays, 0.0, 365.0,
notOnFile, notInTable, notInGraph );
SetWV(m, FreezePoint, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
showStat := FALSE;

1: SetWV(m, month, 0.0, 12.0,
notOnFile, writeInTable, notInGraph );
SetWV(m, day, 1.0, 31.0,
notOnFile, writeInTable, notInGraph );

SetWV(m, hotDayThresh, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
SetWV(m, mTmeanAct, -20.0, 40.0,
notOnFile, writeInTable, isy );
SetWV(m, dTmean, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
SetWV(m, dTmin, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
SetWV(m, nrIceDays, 0.0, 365.0,
notOnFile, notInTable, notInGraph );
SetWV(m, nrFrostDays, 0.0, 365.0,
notOnFile, notInTable, notInGraph );
SetWV(m, nrSummerDays, 0.0, 365.0,
notOnFile, notInTable, notInGraph );
SetWV(m, nrHotDays, 0.0, 365.0,
notOnFile, notInTable, notInGraph );
SetWV(m, FreezePoint, -20.0, 40.0,
notOnFile, notInTable, notInGraph );
showStat := FALSE;

2: SetWV(m, month, 0.0, 12.0,
notOnFile, writeInTable, notInGraph );
SetWV(m, day, 1.0, 31.0,
notOnFile, notInTable, notInGraph );

SetWV(m, hotDayThresh, -20.0, 40.0,
notOnFile, notInTable, isy );
SetWV(m, dTmax, -20.0, 40.0,
notOnFile, writeInTable, isy );
SetWV(m, dTmin, -20.0, 40.0,
notOnFile, writeInTable, isy );
SetWV(m, mTmeanAct, -20.0, 40.0,
notOnFile, notInTable, isy );
SetWV(m, dTmean, -20.0, 40.0,
notOnFile, writeInTable, isy );
SetWV(m, nrIceDays, 0.0, 365.0,
notOnFile, writeInTable, notInGraph );
SetWV(m, nrFrostDays, 0.0, 365.0,
notOnFile, writeInTable, notInGraph );
SetWV(m, nrSummerDays, 0.0, 365.0,
notOnFile, writeInTable, notInGraph );
SetWV(m, nrHotDays, 0.0, 365.0,
notOnFile, writeInTable, notInGraph );
SetWV(m, FreezePoint, -20.0, 40.0,
notOnFile, writeInTable, notInGraph );
showStat := FALSE;
END SetWeatherMonitoring;
ELSE
END(*CASE*);
END SetWeatherMonitoring;

FREEZEPOINT := 0.0;
SHOWSTAT := FALSE;
INSTALL(U);

```

END Weather.

Treibhausklima - Modul GreenClim

IMPLEMENTATION MODULE GreenClim :

Version written for: (1-Pass Modul-2 Implementation)
 MacdETH v2.5.1 (User Interface)
 Dialog Machine V1.0 (Modelling & Simulation environment)
 ModelWorks V1.2

Purpose: Submodel for the greenhouse effect

It is part of the case study model developed in the course "Systemanalyse II (Naturwissenschaftenliche Anwendungen) Teil A. Fischlin, SS 89". The synthesis of the whole model is made by means of the "Looking outward matrix".

List of all submodels:
 K - GreenClim Treibhausklima
 W - Weather Witterung
 P - Primpflod Primäproduzenten
 H - Herbivores Herbivore
 C - Carnivores Karnivore
 D - Detritents Detritenten

Remarks: Systems Ecology version

Implementation and Revisions:

Author	Date	Description
dg	23/05/89	Implementation of module-structure.
dg	21/06/89	Adaptation to student's ID-Matrix.
dg	26/06/89	Final working version
at	03/07/89	Final Implementation of SE model

```

*****
FROM SImbase IMPORT
Model, IntegratiOnMethod, DeclM,
DeclV,
Rectype, Rectip,
StrainType, Tabulation, Graphing, DeclMW,
CurrentType, Currentize, SetIntegratiOnStep,
SetIntim, SetIntIncrval, SetIntegratiOnStep,
TerminalConditionProcedure, InstallTerminalCondition,
StrainIDName, DeclExperiment, SetV, SetW ;

FROM SImbasec IMPORT RunSImbasec, SImrun ;
FROM TabFunc IMPORT TabFunc, DeclTraF, YI, Yie;
FROM SImIntegrate IMPORT Integrate;
    
```

```

(* Model version for many year (always present) *)
(* Model version for 1 year *)
*****
VAR
mForManyYears: Model ; annualEmmLstTable: TabFunc;
bPF, bPFnext, (* CO2 emmLstoma by fossil fuel burning *)
conccO2, (* CO2 concentration in atmosphere *)
(* global temperature reported by this module *)
rcO2max, tSat, t0,deltaCO2, gamma, tau,
conccO2ini1850, conccO2ini1960, annualMean1960 (*model parameters*)
; REAL;
    
```

```

PROCEDURE DeclTable :
CONST
  lModifiable = TRUE ; notModifiable = FALSE ;
VAR x, y : ARRAY[0..7] OF REAL;
BEGIN
  (* Data from Houghton, R.A. & Woodwell, G.M. 1969, Sci. Amer.
  260 (April) 18-26, p.23 Fig.d *)
  x[0] := 1850.0 ; y[0] := 0.05 ;
  x[01] := 1870.0 ; y[01] := 0.1 ;
  x[02] := 1890.0 ; y[02] := 0.2 ;
  x[03] := 1910.0 ; y[03] := 0.6 ;
  x[04] := 1930.0 ; y[04] := 1.4 ;
  x[05] := 1950.0 ; y[05] := 1.8 ;
  x[06] := 1970.0 ; y[06] := 4.0 ;
  x[07] := 1990.0 ; y[07] := 5.5 ;

  DeclTraF := annualEmmLstTable, x, y, g, lModifiable,
  "CO2-Emissionen aus Verbrauch fossiler Brennstoffe",
  curYearIdent, "CO2-emiss" curYearIdent, "Gt C",
  curYearID, curYearID, 0.0, 100.0 ;
    
```

```

END DeclTable;

PROCEDURE AboutModel: BEGIN END AboutModel;
PROCEDURE Input: BEGIN END Input;
PROCEDURE Terminal: BEGIN END Terminal;

PROCEDURE InitLstForManyYears:
BEGIN
  IF CurrentTime() < curYearID THEN
  TO := 1960.0;
  ELSE
  TO := CurrentTime();
  SetV(mForManyYears, bPF, YI(annualEmmLstTable,t0) );
  SetW(mForManyYears, conccO2, 280.0);
  END InitLstForManyYears;

PROCEDURE DynamLstForManyYears:
    
```

```

(* compute only if time in range [curYearID, curYearID] *)
IF CurrentTime() >= curYearID THEN
  bPFnext := (1.0+rcO2max*(tSat-(CurrentTime()-t0)/(tSat))*bPF
  + deltaCO2;
  (* Linear growth, deltaCO2 < 0 - avoid negative bPF *)
  IF bPFnext < 0 THEN bPFnext := 0.0 END;
  END.(tF*);
END DynamLstForManyYears;
    
```

```

PROCEDURE OutputForManyYears:
BEGIN
  conccO2 := conccO2ini1850 + gamma*bPF;
  annualMean := annualMean1960 + (conccO2 - conccO2ini1960)*tau;
  END OutputForManyYears;

PROCEDURE ObjectForManyYears:
BEGIN
  conccO2ini1960 := 310.0; annualMean1960 := 15.0;
  conccO2ini1850 := 290.0;
    
```

```

DeclP (rcO2max, 0.023 - 0.0, 0.2, etc,
"Zwachsrate Verbrauch fossiler Brennstoffe",
"rcO2max", "1/a" );
DeclP (tSat, 2000.0 - 0.0, 4000.0, etc,
"Jahre bis zur Verbrauchssättigung", "tSat", "a" );
DeclP (deltaCO2, 0.0, -1.5, 1.5, etc,
"Änderung des Verbrauchs fossiler Brennstoffe",
"deltaCO2", "Gt C/a" );
DeclP (gamma, 10.0, 0.0, 50.0, etc,
"CO2-Zunahme per Emission", "gamma", "ppm/Gt C/a" );
DeclP (tau, 0.012, 0.0, 1.0, etc,
"Erwärmung per CO2-Zunahme", "tau", "°C/ppm" );

DeclV ( bPF, bPFnext, 5.5, 0.0, 100.0,
"Verbrauch fossiler Brennstoffe", "bPF", "Gt C" );
    
```

END ObjLstForManyYears;

```

*****
(* Model version for 1 year *)
*****
VAR
mForYear: Model ;
    
```

```

PROCEDURE InitLstForYear:
BEGIN
  Integrate(mForManyYears, t0, curYear );
  OutputForManyYears;
  END InitLstForYear;

PROCEDURE DynamLstForYear: BEGIN END DynamLstForYear;
PROCEDURE OutputForYear: BEGIN END OutputForYear;
    
```

```

PROCEDURE ObjectForYear:
BEGIN
  DeclP ( curYear, 1988.0, curYearID, curYearID, etc,
  curYearDescr, curYearIdent, curYearID );
  END ObjLstForYear;
    
```

```

*****
(* Common to both versions *)
*****
    
```

```

PROCEDURE UseGreenhouseClimateModel ( forManyYears: BOOLEAN );
BEGIN
  DeclM ( mForManyYears, discretetime, InitLstForYear,
  Input, OutputForYear, DynamLstForYear,
  Terminal, ObjectForManyYears,
  "Langfristige Entwicklung Treibhausklima", "Global change",
  AboutModel );
  IF forManyYears THEN
  SetSimTime(1900.0, 2050.0);
  ELSE
  DeclM ( mForYear, discretetime, InitLstForYear,
  Input, OutputForYear, DynamLstForYear,
  Terminal, ObjectForYear,
  "Untermodell Jahresklima", "Climate", AboutModel );
  END.(tF*);
  DeclTable;
  END UseGreenhouseClimateModel ;
    
```

```

PROCEDURE SetGreenhouseClimateMonitoring (level: INTEGER);
BEGIN
  CASE level OF
  0: SetW(mForManyYears, bPF, 0.0, 25.0,
  noConFile, notIntrable, notInGraph );
  SetW(mForManyYears, conccO2, 280.0, 1200.0,
  noConFile, notIntrable, notInGraph );
  SetW(mForManyYears, annualMean, annualMeanMin, 25.0,
  noConFile, notIntrable, notInGraph );
  | 1: SetW(mForManyYears, bPF, 0.0, 25.0,
  noConFile, writeIntrable, 1st );
  SetW(mForManyYears, conccO2, 280.0, 1200.0,
  noConFile, writeIntrable, 1st );
  SetW(mForManyYears, annualMean, annualMeanMin, 25.0,
  noConFile, writeIntrable, 1st );
  | 2: SetW(mForManyYears, bPF, 0.0, 25.0,
  noConFile, writeIntrable, 1st );
  SetW(mForManyYears, conccO2, 280.0, 1200.0,
  noConFile, writeIntrable, 1st );
  SetW(mForManyYears, annualMean, annualMeanMin, 25.0,
  noConFile, writeIntrable, 1st );
  
```



```
ELSE  
END(*CASE*);  
END SetGreenhouseClimateMonitoring;
```

```
END GreenhClim.
```

BERICHTE DER FACHGRUPPE SYSTEMÖKOLOGIE
SYSTEMS ECOLOGY REPORTS
ETH ZÜRICH

Nr./No.

- 1 FISCHLIN, A., BLANKE, T., GYALISTRAS, D., BALTENSWEILER, M., NEMECEK, T., ROTH, O. & ULRICH, M. (1991, erw. und korr. Aufl. 1993): Unterrichtsprogramm "Weltmodell2"
- 2 FISCHLIN, A. & ULRICH, M. (1990): Unterrichtsprogramm "Stabilität"
- 3 FISCHLIN, A. & ULRICH, M. (1990): Unterrichtsprogramm "Drosophila"
- 4 ROTH, O. (1990): Maisreife - das Konzept der physiologischen Zeit
- 5 FISCHLIN, A., ROTH, O., BLANKE, T., BUGMANN, H., GYALISTRAS, D. & THOMMEN, F. (1990): Fallstudie interdisziplinäre Modellierung eines terrestrischen Ökosystems unter Einfluss des Treibhauseffektes
- 6 FISCHLIN, A. (1990): On Daisyworlds: The Reconstruction of a Model on the Gaia Hypothesis
- 7 * GYALISTRAS, D. (1990): Implementing a One-Dimensional Energy Balance Climatic Model on a Microcomputer (*out of print*)
- 8 * FISCHLIN, A., & ROTH, O., GYALISTRAS, D., ULRICH, M. UND NEMECEK, T. (1990): ModelWorks - An Interactive Simulation Environment for Personal Computers and Workstations (*out of print*] for new edition see title 14)
- 9 FISCHLIN, A. (1990): Interactive Modeling and Simulation of Environmental Systems on Workstations
- 10 ROTH, O., DERRON, J., FISCHLIN, A., NEMECEK, T. & ULRICH, M. (1992): Implementation and Parameter Adaptation of a Potato Crop Simulation Model Combined with a Soil Water Subsystem
- 11 * NEMECEK, T., FISCHLIN, A., ROTH, O. & DERRON, J. (1993): Quantifying Behaviour Sequences of Winged Aphids on Potato Plants for Virus Epidemic Models
- 12 FISCHLIN, A. (1991): Modellierung und Computersimulationen in den Umweltnaturwissenschaften
- 13 FISCHLIN, A. & BUGMANN, H. (1992): Think Globally, Act Locally! A Small Country Case Study in Reducing Net CO₂ Emissions by Carbon Fixation Policies
- 14 FISCHLIN, A., GYALISTRAS, D., ROTH, O., ULRICH, M., THÖNY, J., NEMECEK, T., BUGMANN, H. & THOMMEN, F. (1994): ModelWorks 2.2 – An Interactive Simulation Environment for Personal Computers and Workstations
- 15 FISCHLIN, A., BUGMANN, H. & GYALISTRAS, D. (1992): Sensitivity of a Forest Ecosystem Model to Climate Parametrization Schemes
- 16 FISCHLIN, A. & BUGMANN, H. (1993): Comparing the Behaviour of Mountainous Forest Succession Models in a Changing Climate
- 17 GYALISTRAS, D., STORCH, H. v., FISCHLIN, A., BENISTON, M. (1994): Linking GCM-Simulated Climatic Changes to Ecosystem Models: Case Studies of Statistical Down-scaling in the Alps
- 18 NEMECEK, T., FISCHLIN, A., DERRON, J. & ROTH, O. (1993): Distance and Direction of Trivial Flights of Aphids in a Potato Field
- 19 PERRUCHOUD, D. & FISCHLIN, A. (1994): The Response of the Carbon Cycle in Undisturbed Forest Ecosystems to Climate Change: A Review of Plant–Soil Models
- 20 THÖNY, J. (1994): Practical considerations on portable Modula 2 code
- 21 THÖNY, J., FISCHLIN, A. & GYALISTRAS, D. (1994): Introducing RASS - The RAMSES Simulation Server

* Out of print

- 22 GYALISTRAS, D. & FISCHLIN, A. (1996): Derivation of climate change scenarios for mountainous ecosystems: A GCM-based method and the case study of Valais, Switzerland
- 23 LÖFFLER, T.J. (1996): How To Write Fast Programs
- 24 LÖFFLER, T.J., FISCHLIN, A., LISCHKE, H. & ULRICH, M. (1996): Benchmark Experiments on Workstations
- 25 FISCHLIN, A., LISCHKE, H. & BUGMANN, H. (1995): The Fate of Forests In a Changing Climate: Model Validation and Simulation Results From the Alps
- 26 LISCHKE, H., LÖFFLER, T.J., FISCHLIN, A. (1996): Calculating temperature dependence over long time periods: Derivation of methods
- 27 LISCHKE, H., LÖFFLER, T.J., FISCHLIN, A. (1996): Calculating temperature dependence over long time periods: A comparison of methods
- 28 LISCHKE, H., LÖFFLER, T.J., FISCHLIN, A. (1996): Aggregation of Individual Trees and Patches in Forest Succession Models: Capturing Variability with Height Structured Random Dispersions
- 29 FISCHLIN, A., BUCHTER, B., MATILE, L., AMMON, K., HEPPELLE, E., LEIFELD, J. & FUHRER, J. (2003): Bestandesaufnahme zum Thema Senken in der Schweiz. Verfasst im Auftrag des BUWAL
- 30 KELLER, D., 2003. *Introduction to the Dialog Machine*, 2nd ed. Price,B (editor of 2nd ed)

Erhältlich bei / Download from

<http://www.ito.umnw.ethz.ch/SysEcol/Reports.html>

Diese Berichte können in gedruckter Form auch bei folgender Adresse zum Selbstkostenpreis bezogen werden /
Order any of the listed reports against printing costs and minimal handling charge from the following address:

SYSTEMS ECOLOGY ETHZ, INSTITUTE OF TERRESTRIAL ECOLOGY GRABENSTRASSE 3, CH-8952 SCHLIEREN/ZURICH, SWITZERLAND
