Univerza v Ljubljani
Fakulteta za gradbeništvo in geodezijo

Nataša Atanasova

Priprava in uporaba ekspertnega predznanja za avtomatizirano modeliranje vodnih ekosistemov

Doktorska disertacija

Mentor: izr. prof. dr. Boris Kompare

Somentor: izr. prof. dr. Sašo Džeroski

Ljubljana, maj 2005

Zahvala

Mentorju, Borisu Komparetu se iskreno zahvaljujem za neprecenljive nasvete, pomoč in podporo, tako pri nastajanju tega dela, kot tudi pri celotnem podiplomskem usposabljanju.

Naloga je rezultat sodelovanja s kolegi z Inštituta Jožef Štefan, Sašom Džeroskim in Ljupčom Todorovskim, od koder se je tudi razvila osnovna ideja za nastanek tega dela. Zato, in za številne diskusije in pomoč, se najlepše zahvaljujem somentorju Sašu Džeroskemu. Posebna zahvala gre Ljupču Todorovskemu za zelo vzpodbudno in prijetno sodelovanje, ter vpeljavo v avtomatizirano modeliranje z *Lagramge-om*.

Del te naloge je nastal v sodelovanju s Fredom Recknagelom, ob študijskem obisku njegove katedre na oddelku *School of Earth & Environmental Sciences, University of Adelaide (Avstralija)*. Fredu Recknagelu se lepo zahvaljujem za konstruktivno sodelovanje, prijetno delovno vzdušje in za posredovane podatke jezera Kasumigaura.

Špeli Rekar se zahvaljujem za posredovanje, interpretacijo in pomoč pri obdelovanju podatkov z Blejskega jezera. Prav tako hvala za razsvetlitev številnih problemov okrog jezera.

Svenu Eriku Jørgensenu hvala za podatke z jezera Glumsø.

Sodelavcem z IZH se zahvaljujem za podporo in mnoge nasvete, še zlasti Mateju Uršiču za delni prevzem mojih pedagoških obveznosti, ter prof. Mitji Rismalu za številne diskusije okrog Blejskega jezera.

Gregorju Petkovšku iskrena hvala za skrbno čitanje rokopisa doktorata in predlagane popravke.

Nazadnje hvala tudi vsem, ki ste mi na kakršenkoli način, bodisi z besedo ali dejanji, pomagali pri izdelavi te naloge in skozi celotni doktorski študij.

Moj doktorski študij je finančno podprla Agencija za raziskovalno dejavnost RS, po pogodbi št. 3311-02-831/433.

Povzetek

Naloga se ukvarja z avtomatiziranim modeliranjem (AM) vodnih ekosistemov. Uporabljena metoda AM (LAGRAMGE) združuje dva osnovna principa modeliranja, t.j. gradnja modelov iz podatkov (empirično), tako kot večina orodij AM in modeliranje z uporabo področnega (teoretičnega) znanja. Združitev teoretičnega in empiričnega pristopa k modeliranju temelji na vpeljavi področnega predznanja v postopek indukcije modelov iz podatkov. Teoretično znanje se upošteva v obliki knjižnice posplošenega znanja iz domene.

Vsebinsko je naloga je razdeljena v dva dela. V prvem delu se ukvarjamo z izdelavo posplošene knjižnice znanja za področje modeliranja vodnih ekosistemov. Natančneje, se zajeto znanje nanaša na modeliranje vodnih ekosistemov z upoštevanjem principa masnih bilanc. Posplošeno znanje o dinamiki sistema je formalizirano preko vpeljave (1) generičnih tipov sistemskih spremenljivk, (2) generičnih osnovnih procesov, ki delujejo na spremenljivke, (3) alternativnih modelov osnovnih procesov in (4) znanja o kombiniranju procesov v model celotnega sistema. Ovrednotili smo splošnost znanja v izdelani knjižnici. Z uporabo predznanja v knjižnici smo zapisali več znanih in uveljavljenih modelov vodnih ekosistemov. Tako smo pokazali (poleg splošnosti zajetega znanja), da ustrezno formalizirano znanje omogoča poenoten modularni pristop tako k 'ročni' gradnji modelov kot tudi avtomatski indukciji modelov iz meritev.

Drugi del naloge se ukvarja z uporabo metode avtomatiziranega modeliranja (LAGRAMGE), ki zdaj vključuje razvito knjižnico znanja, na realnih podatkih. Z uporabo merjenih podatkov in knjižnice smo zgradili modele, ter jih ovrednotili glede na natančnost in razumljivost oz. transparentnost. Obravnavali smo štiri domene: jezero Glumsø, Beneška Laguna, jezero Kasumigaura in Blejsko jezero. Kvaliteta odkritih modelov je odvisna predvsem od (1) znanja zajetega v knjižnici, (2) kvalitete podatkov, (3) kompleksnosti ekosistema in (4) ekspertnega znanja, ki ga vnesemo v postopek odkrivanja modela.

Ključne besede: vodni ekosistem, matematično modeliranje, konceptualno modeliranje, dinamični sistemi, avtomatizirano modeliranje, strojno učenje, domenska knjižnica znanja.

Abstract

This thesis is concerned with automated modelling (AM) of aquatic ecosystems. The method used here integrates the two basic principles of modelling, i.e., empirical or data-driven in theoretical or modelling by using the expert background knowledge. The integration of empirical in theoretical modelling is based on the use of the background knowledge in the procedure of model induction from measured data. The theoretical knowledge that guides the process of model induction includes a knowledge library of generalised knowledge from a specific domain in a task specification of the observed system.

The thesis is divided into two parts. The first part deals with elaboration of knowledge library in the domain of modelling of aquatic ecosystems. The library includes knowledge about food web modeling by following the mass conservation principle. The knowledge is formalized in terms of (1) taxonomy of variable types, (2) basic processes that govern the behavior of aquatic ecosystems, (3) alternative models of the basic processes, and (4) knowledge how to combine models of individual processes into a model of the entire ecosystem. We evaluated the generality of the knowledge in the library through reconstruction of three well-known models of different complexity. Thus, we showed that such formalization of the modelling knowledge provides a solid unifying framework for both handcrafting ecological models as well as their automated induction from measured data.

In the second part we applied the developed library in the AM method on four real world domains. Using the measurements and the background knowledge we constructed models for each domain. The models were evaluated according to their accuracy and transparency. We tackled the following domains: Lake Glumsoe (Danmark), Lagoon of Venice (Italy), Lake Kasumigaura (Japan), and Lake of Bled (Slovenia). The quality of the models is above all dependant on (1) the knowledge in the library, (2) the quality of the measurements, (3) ecosystem complexity, and (4) the expert knowledge introduced in the induction procedure.

Key words: aquatic ecosistems, mathematical modelling, conceptual modelling, dynamic systems, automated modelling, machine learning, domain knowledge library.

Kazalo

Seznam tabel

Seznam slik

Seznam prilog

Priloga A:
Atanasova, N., Todorovski, L., Džeroski, S., Kompare, B. 2005. Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *Ecological Modelling*. Sprejeto za objavo.

Priloga A.1:
Izpis celotne knjižnice znanja o modeliranju vodnih ekosistemov z navadnimi diferencialnimi enačbami.

Priloga B:
Atanasova, N., Todorovski, L., Džeroski, S., Kompare, B. 2005. The use of the expert modelling knowledge in the procedure of automated modelling.

Priloga C:
Atanasova, N., Todorovski, L., Džeroski, S., Recknagel, F., Kompare, B. 2005. Computational Assemblage of Ordinary Differential Equations for Chlorophyll-*a* Using a Lake Process Equation Library and Measured Data of Lake Kasumigaura. In Recknagel, F. (Ed.): Ecological Informatics, 2nd edition. Springer-Verlag 2005. Sprejeto za objavo.

Priloga D:
Atanasova, N., Todorovski, L., Džeroski, S., Kompare, B. 2005. Application of automated model discovery from data and expert knowledge to real world domain: Lake Glumsø, Denmark. *Fifth European Conference on Ecological Modelling (ECEM5)*, 19-23 September 2005. Pushchino, Moscow Region, Russia. Sprejeto.

Priloga E:
Atanasova, N., Todorovski, L., Džeroski, S., Rekar-Remec, Š., Recknagel, F., Kompare, B. 2005. Automated modelling of a food web in Lake Bled using measured data and a library of domain knowledge. *Ecological Modelling*. Sprejeto za objavo.

1 Uvod

Razvoj računalnikov je omogočil hitre in obsežne preračune kompleksnih matematičnih modelov, kamor sodijo tudi ekološki modeli. Kompleksne ekološke modele danes lahko zgradimo po dveh principih. Prvi je deduktivni princip, po katerem na podlagi teoretičnega znanja in opazovanja procesa zasnujemo matematični model s konceptualnimi opisi v obliki matematičnih izrazov. Ker so vsi procesi opisani in sledljivi, takim modelom rečemo transparentni modeli (transparent box, oz. white-box). Drugi je induktivni princip, po katerem matematični model opišemo z neko prenosno funkcijo (preslikavo) med poznanim vhodom in izhodom v/iz opazovanega sistema. Sama prenosna funkcija praviloma nima neposredne veze s fizikalnim, oz. domenskim ozadjem sistema in je običajno le neka regresijska povezava (npr. linearna kombinacija) izhoda z vhodom. Takim modelom rečemo vhodno-izhodni modeli, oz. modeli z nepoznano strukturo (black-box modeli).

Žal se je pokazalo, da lahko deduktivno izpeljani kompleksni modeli zahtevajo preveč podatkov za njihovo umerjanje. Jørgensen (Jørgensen, 1992; Jørgensen, 2002) je celo vpeljal Heisenbergov princip nedoločljivosti na področje ekosistemov in s tem zelo nazorno pokazal, da ne bomo nikoli imeli dovolj opazovanj za natančen opis ekosistema. Izrazna možnost deduktivno izvedenih matematičnih modelov se torej ne povečuje z njihovo kompleksnostjo, pač pa pri neki kompleksnosti doseže svoj optimum (maksimum), nato pa začne zaradi nedoločljivosti oz. nekalibriranosti modela upadati (Constanza in Sklar, 1985). Po drugi strani pa lahko induktivno izvedeni matematični modeli načeloma obravnavajo procese poljubne kompleksnosti, če le uspemo v fazi kalibracije zagotoviti dovolj enolično transformacijo vhoda v izhod modela.

Kompare (1995) je v svoji doktorski tezi uporabil induktivni pristop z uporabo naprednejših orodij za iskanje zakonitosti v podatkih (data mining). Z orodji strojnega učenja je z indukcijo dobil modele, ki so ga navdihnili pri gradnji konceptualnih deduktivnih modelov in na ta način pokazal smiselnost združitve obeh modelirnih principov. Izhajajoč iz teh rezultatov in dejstva, da orodja za avtomatizirano modeliranje, oz. odkrivanje enačb preiskujejo (pre)velik prostor možnih rešitev, sta Todorovski in Džeroski (1997) začela raziskovati možnosti za združitev obeh modelirnih principov, s katero bi omejili velikost tega. Združitev teoretičnega (deduktivnega) in empiričnega (induktivnega) pristopa k modeliranju temelji na vpeljavi domenskega znanja v postopek učenja modelov. Todorovski in Džeroski (1997) sta obravnavo predznanja omogočila z uporabo gramatik za določanje prostora hipotez, kasneje pa sta predlagala zapis predznanja o modeliranju v obliki generičnih procesov (Džeroski in Todorovski, 2003; Langley et al., 2002; Todorovski, 2003).

Ilustrativno bazo predznanja za modeliranje procesov populacijske dinamike predlaga Todorovski (2003), vendar je le-ta precej enostavna in ne ustreza nivoju podrobnosti v tipičnih modelih ekosistemov. Tako je sedaj potreben nov korak

naprej in sicer izdelava, oz. artikulacija in vrednotenje uporabnosti poglobljene baze ekspertnega znanja za eno ali več realnih domen.

Teza se ukvarja z izdelavo take baze oz. knjižnice ekspertnega znanja o vodnih ekosistemih in njeno aplikacijo na realnih merjenih podatkih. Če se opremo na Gruberjevo definicijo (Gruber, 1993) ontologije, gre pravzaprav za izdelavo le-te na področju modeliranja vodnih ekosistemov, saj ima vse potrebne lastnosti, značilne za ontologijo (1) konceptualizacija domenskega znanja na področju modeliranja vodnih ekosistemov (2) njegov zapis v formalizmu, ki vsebuje osnovne gradnike v obliki taksonomije spremenljivk in procesov, ter pravila o tem kako gradimo modele iz teh gradnikov (ta pravila določajo relacije med gradniki) in (3) uporaba ontologije za gradnjo modelov ter izmenjavo znanja na omenjenem področju.

V nalogi je navedena uporabnost izdelane knjižnice znanja (ontologije) ter prednosti v primerjavi z drugimi pristopi k modeliranju, t.j. samo dedukcijskega ali samo indukcijskega izhodišča.

1.1 Pregled stanja na področju modeliranja naravnih procesov

Začetki modeliranja, t.j. opisa naravnih procesov v ekosistemu z matematičnimi sredstvi segajo v 20-ta leta 20. stoletja. Eden prvih takih modelov v vodarstvu je npr. Streeter-Phelps model (Streeter in Phelps, 1925), ki napoveduje koncentracijo kisika vzdolž vodotoka, v katerega se izpušča (neprečiščena) odpadna voda. Drugi zelo popularni model v ekologiji je model Lotka-Voltera (Lotka, 1924; Volterra, 1931), ki opisuje dinamiko populacij plenilca in plena. V 60-ih se je pojavila množica enostavnejših modelov za predikcijo evtrofikacije, ki so bazirali bolj na empiriki, kot pa na poglobljenem razumevanju procesov. Eden prvih je Vollenweiderjev polempirični model (Vollenweider, 1968), ki vsebuje eno samo enačbo za določitev povprečne letne koncentracije fosforja v jezeru. Model sloni na statistično določeni regresijski enačbi z empiričnimi nastavki. Zaradi opisa jezera kot popolnoma premešanega homogeniziranega telesa (reaktorja), tak model imenujemo eno-oddelčni (one compartment) model. Nadaljni razvoj modelov je šel v smeri dinamičnega (časovno odvisnega) modeliranja več odvisnih spremenljivk oz. spremenljivk stanja, kakor tudi upoštevanja slojevitosti jezera, ter izmenjave hranil s sedimentom (multi compartment). O'Melia (O'Melia, 1972), Imboden (Imboden, 1974) in Snodgrass (Snodgrass, 1974) so podali smernice za simulacijo takih sistemov. Veliko konceptov ekološkega modeliranja in formulacij naravnih procesov podajajo npr. (Jørgensen in Bendoricchio, 2001; Patten in Jørgensen, 1995; Chapra, 199; Bowie et al., 1985; DeAngelis, 1992; Andersen, 1997; Chen in Orlob, 1975) itd.

Prehod od oddelčnih sistemov k prostorskim sistemom je bil bolj težaven, saj se matematični opisi zapletejo in zahtevajo namesto navadnih diferencialnih enačb parcialne diferencialne enačbe. S čisto teoretičnega vidika to sicer ne predstavlja bistvenih težav, v praksi pa je reševanje parcialnih diferencialnih enačb vezano na numerične postopke, ki sami po sebi ne zagotavljajo natančne, oz. inženirsko sprejemljive rešitve. Primer enodimenzionalnega (1D) dinamičnega modela je model

reke QUAL2E (Roesner et al., 1991), kjer je dimenzija v vzdolžni smeri, oz. model jezera, kjer je dimenzija v navpični smeri (Center for Water Research (CWR), 2003CWR). Kombinacija jezera in reke, oz. rečna akumulacija tako zahteva vsaj dvodimenzionalen (2D) model, t.j. v vzdolžni in navpični smeri, če ne celo kompletno tri-dimenzionalnega (3D), torej tudi v prečni horizontalni smeri. Takih 3D modelov jezera je malo, pa še ti se večinoma bolj posvečajo hidrodinamiki, kot pa kvaliteti, oz. opisovanju procesov evtrofikacije (Imberger in Ivey, 1991; Četina, 1988; Rajar in Četina, 1997; Žagar et al., 2001; Rismal et al., 1997). Poleg velike numerične zahtevnosti teh modelov (potrebni so računalniki zadnje generacije) se pojavlja še problem kalibracije teh modelov, saj praviloma primanjkuje merskih podatkov, ki bi zadostili pogojem uspešne kalibracije. Tako najbolj kompleksni modeli praviloma niso dovolj ovrednoteni in dovolj zanesljivi za namen, za katerega so sicer bili zgrajeni.

Iz izkušenj z modeliranjem naravnih sistemov, predvsem vremena, je kmalu postalo jasno, da tudi enostavni matematični modeli s tremi diferencialnimi enačbami lahko opisujejo neverjetno kompleksne, kaotične vzorce, t.j. periodična ali neperiodična nihanja in katastrofe, t.j. nezvezne prehode, ki so jih opisali (Steward, 1989), (Gleick, 1991), (Bossel, 1994) in drugi.

Vsled navedenih pomanjkljivosti preveč kompleksnih dedukcionistično razvitih matematičnih modelov je primerno, da kljub poglobljenemu razumevanju osnovnih (enotnih) in sinergističnih procesov matematični model čimbolj poenostavimo. Tudi iz teorije kaosa (Steward, 1989; Gleick, 1991) sledi, da je večina naravnih sistemov v neki ravnovesni točki relativno stabilnih in ne izkazuje kaotičnega obnašanja – pač pa se limitno vrača v ravnotežni sistem k t.i. atraktorjem. V takem stanju se kompleksen konceptualni matematični opis lahko reducira na bistveno bolj enostavnega. Kompare (1995) je v svojih raziskavah za Beneško laguno namesto sistema 11 diferencialnih in algebrskih enačb s strojnim učenjem iz podatkov dobil eno samo diferencialno enačbo, ki se je navzven obnašala povsem primerljivo kot omenjeni sistem 11-ih enačb. Omenjeno enačbo je odkril s sistemom za odkrivanje enačb GOLDHORN (Križman, 1998).

Odkrivanje enačb je podpodročje avtomatskega modeliranja, ki se ukvarja z učenjem algebrskih (Langley et al., 1987; Kokar, 1986; Zembovich in Zytkow, 1992; Washio in Motoda, 1997) ali navadnih diferencialnih enačb (Todorovski, 1993; Džeroski in Todorovski, 1995; Todorovski, 1998; Todorovski in Džeroski, 1997). Za analizo ekoloških podatkov oz. empirično modeliranje ekosistemov se pogosto uporabljajo tudi druge metode strojnega učenja, kot so npr. indukcija odločitvenih dreves (Quinlan, 1986), regresijskih dreves (Breiman et al., 1984; Quinlan, 1992; Quinlan, 1993) ali pa optimizacija z genetskimi algoritmi. S temi metodami so bili narejeni pomembni poskusi napr. na področju čistilnih naprav za odpadno vodo (Belanche et al., 1999; Comas et al., 2001; Roda et al., 1999; Sanchez et al., 1997; Atanasova in Kompare, 2002b; Atanasova in Kompare, 2002a; Atanasova in Kompare, 2002c) ali pa na področju optimizacije vodooskrbnih sistemov z genetskimi algoritmi (Steinman et al., 2001).

Za predlagano delo so najbolj relevantne raziskave na področju združitve teoretičnega (deduktivnega) in empiričnega (induktivnega) pristopa k modeliranju. Le-to temelji na uporabi predznanja. Gre za vpeljavo ekspertnega oz. domenskega znanja v postopek učenja modelov. Todorovski in Džeroski (Todorovski in Džeroski, 1997) sta obravnavo predznanja omogočila z uporabo gramatik za določanje prostora hipotez. Tako je bilo možno sistemu LAGRAMGE z gramatiko podati pričakovane konstrukte enačb – npr. zgradbo Monodovega člena. S tem orodjem je bil uspešno odkrit model cvetenja alg v danskem jezeru Glumsø (Todorovski et al., 1998).

Pred kratkim sta Džeroski in Todorovski predlagala zapis predznanja o modeliranju v obliki generičnih procesov (Džeroski in Todorovski, 2003; Todorovski, 2003). Predznanje na določenem problemskem področju podamo sistemu LAGRAMGE 2.0 v obliki knjižnice generičnih procesov, osnovnih gradnikov za modeliranje le-teh ter načinih za sestavljanje celotnih modelov iz omenjenih sestavin. Kot primer poda Todorovski (Todorovski, 2003) bazo predznanja za modeliranje procesov populacijske dinamike. Žal je ta precej enostavna in ne ustreza nivoju podrobnosti v tipičnih modelih ekosistemov. Logičen in potreben naslednji korak, oz. odprta naloga je izgradnja tovrstne knjižnice za izbrano domeno, npr. evtrofikacije jezera ali delovanja čistilne naprave za odpadne vode, ki bi bila zadosti zajetna in kompleksna, da bi omogočila gradnjo podrobnih in natančnih modelov realnih ekosistemov.

1.2 Pregled orodij za modeliranje naravnih procesov

V osnovi lahko matematični model poljubnega sistema zapišemo z matematičnimi enačbami, le te pa nato zakodiramo v primeren programski jezik in takšen program rešimo z računalniško simulacijo. Takšni so tudi bili začetni pristopi k matematičnemu modeliranju. Že kmalu pa se je izkazalo, da je mogoče določene dele programske kode posplošiti, oz. izdelati modelirna orodja na višjem nivoju.

Osnovni namen specifičnih ali generalnih modelirnih orodij na višjem nivoju je približati modeliranje in simulacijo širši strokovni javnosti, ki se ukvarja z ekologijo. Večina teh orodij danes vsebuje ustrezne grafične vmesnike, tako da se uporabnik ne ukvarja z enačbami in njihovim reševanjem ter ne z grafičnim vnosom in prikazom rezultatov. Orodja lahko grupiramo na orodja za konceptualno modeliranje, kvalitativno modeliranje ali t.i. blokovno gradnjo modelov, avtomatizirano konstrukcijo modelov in orodja umetne inteligence.

Izmed množice orodij za konceptualno modeliranje velja omeniti STELLo (isee systems, 2004), LAKE (Mahler in Salomonsen, 1992), AQUASIM (Reichart, 1998), SIMILE (Simulistics, 2005) idr. Detaljnejši pregled orodij za modeliranje in simulacijo je podan tudi na spletu (Rizzoli, 2005). Uporabnik zgradi svoj model preko grafičnega vmesnika, nato pa izvede simulacijo.

Bistvo orodij za kvalitativno modeliranje je pomoč uporabniku pri gradnji matematično pravilnih modelov. Simulacija ni primarni cilj. Tako orodje je recimo ECOBAS Modelling Assistant Tool (EMA) (Benz in Hoch, 1997; Benz et al., 2001; Benz in Knorrenschild, 1997; Benz in Voigt, 1996). Orodje sestavlja sintaksa oz.

jezik, ki omogoča konsistentno formuliranje modelov ter podatkovna baza (ki jo je mogoče nadgrajevati) s predhodno definiranimi modeli.

Pomembne korake na področju avtomatske konstrukcije modelov je naredil Muetzelfeld s sodelavci (Muetzelfeldt et al., 1989; Robertson et. al., 1991). Sistem ECOLOGIC obsega znanje o ekoloških procesih in modelih, do katerega dostopamo preko uporabniškega vmesnika. Slednji uporabniku omogoča, da z odgovorom na vrsto vprašanj izbere ustrezno strukturo modela, vendar se pa ne ukvarja s problematiko kalibriranja parametrov.

Na področju umetne inteligence velja omeniti programski paket WEKA (Witten in Franck, 1999), ki vsebuje večino popularnih algoritmov (različnih avtorjev) strojnega učenja, ne vsebuje pa algoritmov strojnega odkrivanja enačb. Od sistemov za odkrivanje enačb naj omnenim naslednje: Lagrange (Džeroski in Todorovski, 1993; Džeroski in Todorovski, 1995), GoldHorn (Križman, 1998), LAGRAMGE (Todorovski in Džeroski, 1997) in LAGRAMGE 2.0 (Todorovski, 2003). Nobeno izmed navedenih orodij, razen LAGRAMGE-a 2.0 in delno LAGRAMGE-a ne vključuje eksplicitne kombinacije deduktivnega in induktivnega pristopa k modeliranju.

Prispevek pričujočega dela bi torej bil konstrukcija domenske knjižnice za izbrano področje (vodni ekosistemi) in prikaz delovanja te knjižnice v kontekstu programa LAGRAMGE 2.0 na izbranih realnih primerih.

1.3 Namen teze

Naloga ima tri poglavitne cilje:

- Prvi cilj je formalizirati znanje na področju modeliranja jezer, ki se lahko uporablja (1) v teoretične namene, za pomoč k gradnji matematično pravilnih modelov in (2) v postopku avtomatskega modeliranja z orodjem LAGRAMGE 2.0. S tem bo omogočen enotni modularni pristop k gradnji tovrstnih modelov.
- Drugi cilj je prenesti princip združitve empiričnega in teoretičnega znanja na realne primere s področja modeliranja jezer. To pomeni pokazati, da se s tem principom iz merjenih podatkov lahko zgenerirajo različno kompleksni modeli, ki dovolj natančno opisujejo domeno in so istočasno razumljivi in sprejemljivi ekspertom (za razliko od black-box modelov).
- Tretji cilj je približati modeliranje in simulacijo širši strokovni javnosti, ki se ukvarja z ekologijo. Ta cilj bo izpolnjen s tem, da bo omogočena gradnja pravilnih modelov tudi ekspertom, ki jim matematično modeliranje ni primarna domena.

1.4 Prispevki doktorata

Prispevki teze se nanašajo na področje ekološkega modeliranja. Predstavljena je uporaba novega pristopa k modeliranju, ki združuje tako teoretično kot empirično znanje v postopek gradnje modelov. Teza ima tri poglavitne originalne prispevke:

- Izdelava knjižnice ekspertnega domenskega znanja za ekološko modeliranje vodnih ekosistemov. V njej je zbrano in formalizirano (v sintaksi procesnih

modelov) znanje o modeliranju vodnih ekosistemov. To obsega popis osnovnih/generičnih ekoloških procesov (kot so procesi evtrofikacije, npr. dotok hranil in njihovo kroženje v sistemu, in populacijske dinamike, npr. rast, odmiranje, plenilstvo) v vodnih sistemih. Vsebuje tudi tipične gradnike ekoloških modelov, ki ustrezajo posameznim procesom (npr. eksponentna ali logistična rast populacije).
- Evalvacija splošnosti predznanja zajetega v knjižnici. Posplošeno znanje na osnovi procesov, oz. posameznih gradnikov za modeliranje le-teh, omogoča poenoten modularni pristop k gradnji modelov različnih vodnih ekosistemov. Z uporabo predznanja v knjižnici smo zapisali več znanih in uveljavljenih modelov vodnih ekosistemov, od zelo enostavnih, kot je Vollenweider-jev model (Vollenweider, 1968), do zmerno kompleksnih (Imboden, 1974) in razmeroma kompleksnih modelov kot je model SALMO (Bendorf, 1979; Recknagel, 1980).
- Evalvacija knjižnice v kontekstu modeliranja realnih vodnih ekosistemov iz merjenih podatkov in domenskega predznanja. Z omenjenim pristopom k modeliranju smo zgradili uporabne modele naslednjih vodnih ekosistemov: Beneška laguna, Italija, Blejsko jezero, Slovenija, jezero Glumsø, Danska in jezero Kasumigaura, Japonska.

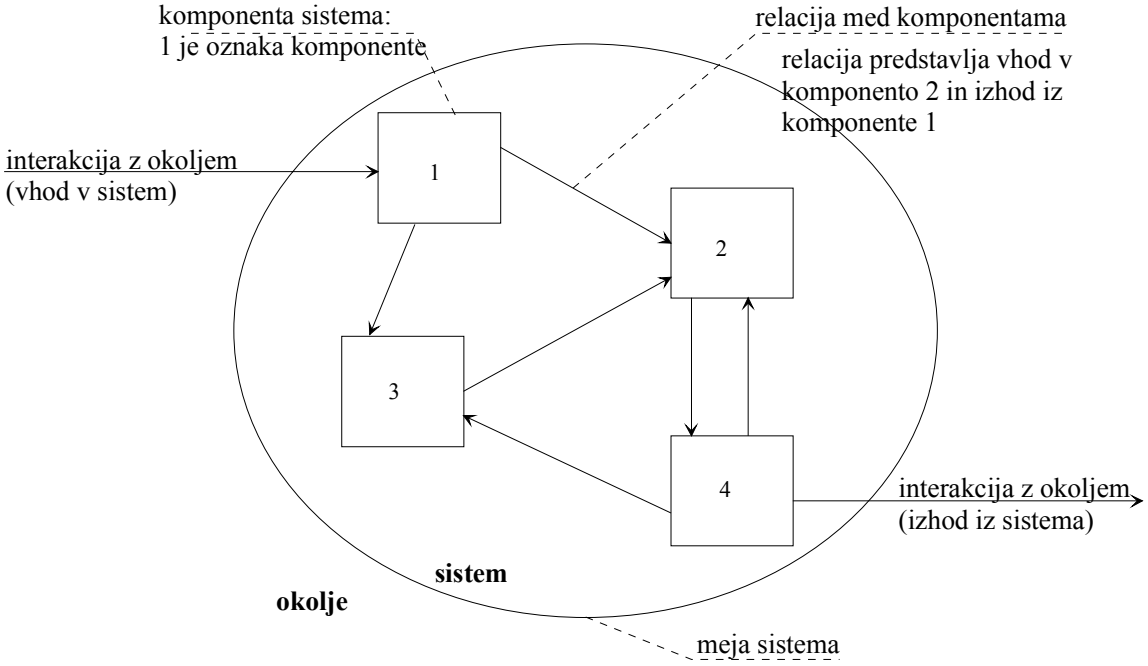
Kot dodatna prispevka doktorata lahko štejemo:

- Izčrpno in konsistentno bazo podatkov za Blejsko jezero. Trenutni podatki meritev kakovostnih spremenljivk in pretokov so razdrobljeni po raznih inštitucijah, podvojeni ali izgubljeni, predvsem pa nepreverjeni. V namen uporabe merjenih podatkov za avtomatizirano ekološko modeliranje smo izdelali izčrpno in preverjeno podatkovno bazo za Blejsko jezero.
- Povečanje zanimanja za gradnjo modelov, še posebej pa avtomatizirano indukcijo modelov na področju ekološkega modeliranja in modeliranja na sploh.

2 Izhodišča in obstoječe metode

2.1 Teoretične osnove za konceptualno modeliranje

Ena najbolj splošnih definicij sistema pravi, da je sistem množica elementov, ki imajo medsebojne relacije in relacije z okoljem (Bertalanffy, 1972). Sistem ima neko notranjo strukturo, ki je definirana z različnimi komponentami. Komponente so povezane z relacijami, ki običajno pomenijo izmenjavo snovi, energije in informacije. Za komponento, ki to relacijo sprejema, relacija pomeni vhod (v komponento), za komponento, iz katere relacija izhaja, pa ta relacija pomeni izhod (iz komponente). Določene komponente imajo relacije tudi z okoljem. Te relacije predstavljajo interakcijo sistema (kot celote in ne posameznih komponent) z okoljem. Relacijam iz okolja, ki imajo vpliv na posamezne komponente sistema, pravimo vhodi v sistem, tistim relacijam sistema, ki pa vplivajo na okolje, pravimo izhodi iz sistema (Slika 1).



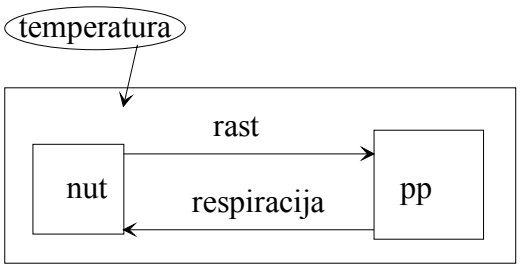
Slika 1: Shema sistema z notranjo strukturo (Strmčnik, 1998)

Zgornjo definicijo lahko uporabimo kot prvi korak pri izdelavi ekološkega modela, t.j. konceptualizacijo ekosistema, ki ga želimo modelirati. V ekološki terminologiji pravimo komponentam *sistemske spremenljivke ali spremenljivke stanja*. Opisujejo bistvene značilnosti in obnašanje sistema ter so funkcije časa in prostora. Poleg spremenljivk stanja poznamo tudi *neodvisne spremenljivke ali gonilne funkcije*, s katerimi predstavljamo okolje sistema. Te spremenljivke so vhod v sistem. Primeri gonilnih funkcij so temperatura, svetloba, prihajajoča hranila v sistem itd. Relacije v zgornji definiciji predstavljajo ekološki procesi izmenjave in transformacije snovi v sistemu, ki jim bomo v nadaljevanju rekli samo *procesi*. Če nas bolj zanima delovanje, oz. stanje sistema, kot pa njegovi izhodi, potem bolj kot izhodom iz

sistema pozornost namenjamo vrednostim nekaterih ključnih spremenljivk stanja (Beck, 1983).

Konceptualizacija vodnega ekosistema vključuje dva osnovna koraka: (1) fizikalno razdelitev vodnega sistema na diskretne elemente in (2) izbiro in razdelitev biotskih komponent glede na njihove vloge v vodnem ekosistemu (Beck, 1983). Konceptualni model sistema lahko zapišemo z matematičnimi izrazi in dobimo matematični model. Eden izmed načinov za matematično formulacijo je implementacija zakona o ohranitvi mase. S tem zakonom formuliramo model, ki opisuje spremembo mase posamezne spremenljivke stanja v času. Če želimo spremembo opisati kontinuirno, t.j. časovno dinamično, uporabljamo navadne diferencialne enačbe. Ker bomo v nadaljevanju opisovali vodne ekosisteme, bomo namesto mase lahko uporabili volumske koncentracije – v primeru, da je volumen konstanten (kar je večinoma res pri nizkih koncentracijah), pišemo masno bilanco kar samo s spremembami koncentracij.

Nazorno lahko postopek od konceptualizacije do matematične formulacije prikažemo na enostavnem primeru dinamike primarnega producenta (*pp*) in anorganskega hranila (*nut*). Naš sistem sestavljata dve spremenljivki stanja (*pp* in *nut*), ena gonilna funkcija oz. neodvisna spremenljivka (temperatura) ter procesi, ki delujejo in vplivajo na maso, oz. koncentracijo obeh spremenljivk. Konceptualno sistem prikazuje Slika 2. Proces *rast* povezuje obe spremenljivki stanja in je usmerjen proti *pp*. Primarni producent (*pp*) se hrani s hranilom (*nut*) in zaradi tega njegova koncentracija, oz. masa narašča. Torej proces *rast* vpliva na naraščanje mase *pp*. Nasprotno se koncentracija hranila *nut* manjša zaradi rasti *pp*. Zaradi procesa *respiracija*, ki je usmerjen proti hranilu *nut*, se masa *pp* manjša, masa hranila *nut* pa povečuje. Zunanja spremenljivka *temperatura* vpliva na procesa rasti in respiracije.



Slika 2: Konceptualni model dinamike primarnega producenta (*pp*) in anorganskega hranila (*nut*)

Z uporabo diferencialnih enačb lahko masne bilance obeh spremenljivk stanja zapišemo kot prikazujeta enačbi (1) in (2), kjer smo časovni diferencial d/dt pisali s črtico nad spremenljivko:

$$\mathrm{d(pp)/dt} = pp' = \mathrm{rast\text{-}respiracija} \tag{1}$$

$$\mathrm{d(nut)/dt} = nut' = -\mathrm{k1} \cdot \mathrm{rast} + \mathrm{k2} \cdot \mathrm{respiracija} \tag{2}$$

Konstanti k1 in k2 v drugi enačbi pomenita pretvorbene faktorje iz biomase (pp) v hranilo (nut), oz. stehiometrijsko razmerje med biomaso in anorganskim hranilom.

Matematična formulacija procesov

Večino ekoloških procesov lahko na podlagi teorije in/ali eksperimentov formuliramo s številnimi modeli. Rast primarnega producenta lahko opišemo npr. z eksponentnim modelom (3) ali pa z modelom, ki upošteva vpliv temperature in koncentracij anorganskih hranil (omejitveni model) (4).

$$rast = \mu \cdot pp \tag{3}$$

$$rast = \mu \cdot f1(T) \cdot f2(nut) \cdot pp \tag{4}$$

kjer je μ hitrost rasti *pp* [1/čas] pri optimalnih pogojih, f1(*T*) funkcija vpliva temperature na rast in f2(*nut*) omejitvena funkcija za rast *pp* zaradi koncentracije hranila *nut*.

V našem primeru bomo uporabili Arrheniusov temperaturni model (5) in Monodov model (Monod, 1949) za omejitev rasti zaradi koncentracije hranil (6).

$$f1(T) = \Theta^{(T-T_{ref})} \tag{5}$$

$$f2(nut) = \frac{nut}{k3 + nut} \tag{6}$$

Če je rast primarnega producenta omejena z več kot enim hranilom, lahko skupni vpliv vseh hranil izrazimo kot produkt omejitvenih funkcij koncentracij posameznega hranila (enačba 7):

$$f(C,N,P) = f(C) \cdot f(N) \cdot f(P) = \frac{C}{k_C + C} \cdot \frac{N}{k_N + N} \cdot \frac{P}{k_P + P} \tag{7}$$

Proces respiracije lahko formuliramo s kinetiko prvega reda (8):

$$respiracija = -k4 \cdot PP \tag{8}$$

Pri teoretičnem, oz. dedukcionističnem pristopu k formulaciji modela se ponavadi ekspert sam odloča, katera formulacija procesov je najprimernejša za posamezni primer. Če za proces *rast* izberemo omejitveni model (4), kot omejitveno funkcijo hranila pa izberemo enačbo (6), dobimo sledečo matematično formulacijo (9) in (10) konceptualnega modela (Slika 2):

$$pp' = \mu \cdot \Theta^{(T-T_{ref})} \cdot \frac{nut}{k3 + nut} \cdot pp - k4 \cdot pp \tag{9}$$

$$nut' = -k1 \cdot \mu \cdot \Theta^{(T-T_{ref})} \cdot \frac{nut}{k3 + nut} \cdot pp + k2 \cdot k4 \cdot pp \tag{10}$$

Koeficienti k1, k2, k3 in k4 so parametri modela, ki jih je potrebno oceniti oz. umeriti glede na podane meritve, če le-te obstajajo.

S tem enostavnim primerom smo načelno prikazali postopek izdelave matematičnega modela z uporabo teoretičnega (pred)znanja. V nadaljevanju prikažemo, kako to (pred)znanje uporabimo v postopku avtomatskega odkrivanja enačb, oz. sistemov.

2.2 Avtomatizirano odkrivanje diferencialnih enačb z uporabo predznanja

2.2.1 Lagramge

LAGRAMGE (Todorovski in Džeroski, 1997) odkriva model z eno diferencialno enačbo, oblike $v_d' = E$, kjer je v_d' časovni odvod spremenljivke v_d, E pa izraz, ki je izpeljan iz t.i. gramatike G. Gramatiko določa uporabnik in predstavlja prostor modelov oz. možnih struktur, ki jih model lahko zavzame.

Vsaka struktura modela v gramatiki vsebuje konstantne parametre, ki jih LAGRAMGE določa tako, da ima model najmanjšo napako glede na podane meritve. Ali drugače, LAGRAMGE izvaja optimizacijo (kalibracijo) vsake izmed mogočih struktur modela v gramatiki. Kvaliteta dobljene enačbe se nato vrednoti s funkcijo srednje vrednosti vsote kvadratov napake, MSE (mean squared error).

$$MSE = \frac{\sum_{i=1}^{m} (v_d(i) - \tilde{v}_d(i))^2}{m} \tag{11}$$

kjer je $v_d(i)$ merjena vrednost spremenljivke v_d v točki i, $\tilde{v}_d(i)$ pa preračunana vednost spremenljivke v_d v točki i s simulacijo odkrite enačbe oblike $\dot{v}_d = E$ in m je število meritev.

Dodatno vsebuje LAGRAMGE še eno funkcijo za vrednotenje dobljenih modelov. Funkcija MDL upošteva kompleksnost dobljene enačbe (12).

$$MDL(v'_d = E) = MSE(v'_d = E) + \frac{l(E)}{10 \cdot l_{\max}} \cdot MSE(v'_d = E_0) \tag{12}$$

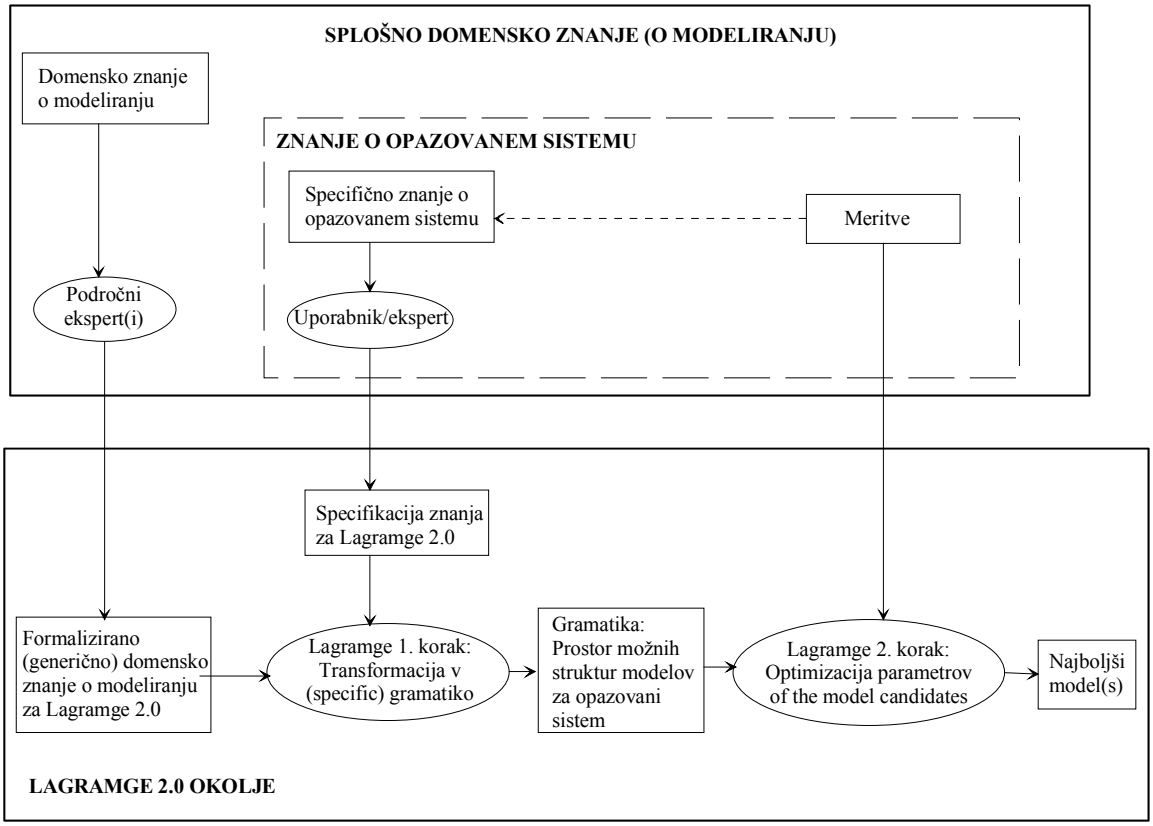
kjer je $l(E)$ dolžina izraza (izražena v številu členov), l_{max} je maksimalna dolžina izraza, ki ga lahko izpeljemo iz gramatike in E_0 najenostavnejši izraz v gramatiki, parameter 10 je izbran na podlagi izkušenj. Drugi člen v MDL funkciji poveča napako MSE glede na dolžino enačbe, oz. daljša enačba bo imela večjo napako. Torej, ta funkcija preferira krajše enačbe (princip Occhamove britve – Occham's razor, William of Ockham, ca.1285-1349).

2.2.2 Lagramge 2.0

Obravnavo predznanja v procesu odkrivanja enačb, kot ga pozna LAGRAMGE, sta Džeroski in Todorovski nadgradila v naslednji verziji LAGRAMGE-a (LAGRAMGE 2.0). Predlagala sta zapis predznanja o modeliranju v obliki generičnih procesov (Džeroski in Todorovski, 2003; Todorovski, 2003). Predznanje na določenem problemskem področju podamo sistemu LAGRAMGE 2.0 v obliki knjižnice generičnih procesov, osnovnih gradnikov za modeliranje le-teh ter načinih za sestavljanje celotnih modelov iz omenjenih sestavin.

Postopek avtomatskega modeliranja (AM) z LAGRAMGE-om 2.0 kaže Slika 3. Domensko znanje, v našem primeru je to splošno-veljavno (generično) znanje o modeliranju prehranjevalnih mrež v jezerih, je zbrano v generični knjižnici znanja (Domain specific modelling knowledge). To generično knjižnico napiše(-jo) izbrani domenski ekspert(-i) v skladu s sintakso jezika, ki jo zahteva LAGRAMGE 2.0 predprocesor za izdelavo knjižnice. To knjižnico sedaj lahko uporabljajo tudi drugi uporabniki, ki niso nujno eksperti na domenskem področju. Vendar pa je za artikulacijo nekega bolj specifičnega sistema (modela) potrebno tudi bolj natančno poznavanje le-tega sistema, tako da v končni konsekvenci definicija modela le ne more biti prepuščena povsem nepoučenim. Bolj specifično (ekspertno) znanje o določenem sistemu, ki ga želimo modelirati, poda uporabnik (strokovnjak) v specifikaciji opazovanega sistema (task specification). Tu je vključena specifikacija opazovanih spremenljivk stanja in procesov, ki so (po mnenju strokovnjaka) relevantni za opis opazovanega ekosistema. Na ta način iz generične knjižnice ustvarimo specificirano, oz. specifično. Ta dva koraka sta bila narejena peš, t.j. s strani eksperta in uporabnika. Naslednji korak je avtomatska (LAGRAMGE 2.0) transformacija generičnega ekspertnega znanja iz knjižnice ob upoštevanju ekspertnega znanja o specifičnem sistemu) v gramatiko, ki določa prostor vseh modelov (različnih struktur), ki ustrezajo ekspertovemu opisu. To je prikazano na levi strani, Slika 3.

Ko je gramatika zgrajena, LAGRAMGE 2.0 prične z iskanjem najboljšega modela in sicer tako, da gre vsak izmed modelov skozi postopek nelinearne optimizacije parametrov, glede na podane meritve. Program vrne niz najboljših modelov, t.j takih, ki se meritvam najbolj približajo. To je ilustrirano na desni strani, Slika 3.



Slika 3: Princip avtomatskega modeliranja (AM) z integracijo področnega znanja v proces odkrivanja enačb s programom LAGRAMGE 2.0 (modificirano po Todorovski, 2003).

2.3 Formalizacija domenskega znanja za uporabo v AM

V tem poglavju podajamo grobi opis formalizma za zapis domenskega znanja v knjižnico generičnih procesov za uporabo v AM. Opis bomo podprli z demonstracijo na enostavnem primeru. Formalizem je razvil in natančneje opisal Todorovski (2003). Podpira modeliranje z navadnimi diferencialnimi enačbami z upoštevanjem masnih bilanc. Koncept takega modeliranja smo pokazali v poglavju 2.1 (glej tudi npr. Jørgensen in Bendoricchio, 2001; DeAngelis, 1992; Chapra, 1997; Orlob et al., 1983 itd.). Z razvitim formalizmom smo generalizirano zapisali znanje o modeliranju prehranjevalnih mrež v vodnih ekosistemih. To pomeni, da so deklarirane spremenljivke in procesi v knjižnici generični. Z ustrezno specifikacijo opazovanega sistema lahko iz knjižnice zgeneriramo različno kompleksne modele, t.j. modele ki vsebujejo natanko toliko spremenljivk in procesov, kolikor jih definiramo v specifikaciji opazovanega sistema. Torej, formalizem za zapis domenskega znanja v knjižnico jezerskega ekosistema vsebuje:

- Deklaracijo tipov spremenljivk
- Deklaracijo procesnih razredov, ki opisujejo dogajanje v vodnih ekosistemih.
- Različne formulacije procesnih razredov, ki predstavljajo procesne podrazrede in
- Kombinatorne sheme osnovnih procesov, ki ponazarjajo masne bilance sistemskih spremenljivk.

Taksonomija tipov spremenljivk obsega odvisne (sistemske) in neodvisne spremenljivke. Poleg osnovnih tipov formalizem podpira tudi deklaracijo podtipov. Deklaracija tipov spremenljivk je nazorno prikazana na primeru v nadaljevanju tega poglavja. Deklarirani tipi spremenljivk nastopajo v formulacijah procesnih razredov.

Procesni razredi predstavljajo relacije oz. procese (glej definicijo ekosistema) med spremenljivkami, oz. komponentami sistema (Slika 1). Procesni razredi torej vplivajo na spremenljivke v sistemu. Tu se izkaže uporabnost deklariranja podtipov osnovnih tipov spremenljivk. Če namreč nek proces deluje (vpliva) na nek osnovni tip spremenljivke, se bo njegov vpliv prenesel tudi na podtip te spremenljivke. V poglavju 1.1 smo videli, da ima lahko proces (oz. procesni razred) več možnih formulacij. Te formulacije podajamo kot podrazrede procesnih razredov. Procesni razred ima toliko podrazredov, kolikor modelov (formulacij) obstaja (oz. jih poznamo) za opis tega procesa. Deklaracijo procesnih razredov prikazuje Tabela 1.

Tabela 1: Deklaracija procesnih razredov v knjižnici

```
#Deklaracija procesnega razreda
process class ime_procesa(tip_spremenljivke1 ime1, tip_spremenljivke2 ime2)
        #Deklaracija podrazredov, t.j. različnih formulacij procesnega razreda

        process class ime1() is ime_procesa
                expression formulacija1
        process class ime2() is ime_procesa
                expression formulacija2
```

Tabela 1 podaja deklaracijo procesa, ki ima dva argumenta (spremenljivka 1 in spremenljivka 2), oz. ga formuliramo z dvema spremenljivkama, ter dve različni formulaciji (podrazredi). Pri definiciji argumentov podajamo tipe spremenljivk, na katere proces deluje, oz so povezane s procesom in s katerimi lahko ta proces formuliramo. Če ima spremenljivka podtip, potem bo proces deloval tudi na podtip(e) te spremenljivke. Formulacijo procesa zapišemo za besedo *expression*. Proces ima lahko toliko podrazredov, kolikor je različnih modelov, oz. formulacij za ta proces. V formulaciji procesa nastopajo argumenti (tipi spremenljivk) in konstante. Konstante (parametre) zapišemo z besedo *const(ime_konstante, spodnja_meja, začetna_vrednost, zgornja_meja)*. Generične konstante se pozneje umerjajo (specificirajo) glede na meritve v postopku optimizacije.

V svojih formulacijah lahko procesni razredi vsebujejo tudi funkcijske razrede. Deklaracija funkcij je zelo podobna deklaraciji procesnih razredov. Uporabnost funkcij bomo pokazali na primeru v nadaljevanju tega poglavja. Spodnje vrstice prikazujejo deklaracijo funkcijskega razreda. Edina sintaktična razlika med deklaracijo procesnih in funkcijskih razredov je ta, da ključno besedo *proces class* zamenjamo z besedo *function class* (Tabela 2).

Tabela 2: Deklaracija funkcijskih razredov v knjižnici

```
1:  function class ime_funkcije((tip_spremenljivke1 ime, tip_spremenljivke2
2:  ime2)
3:          function class ime1() is ime_funkcije
4:                  expression formulacija1
5:          function class ime2() is ime_funkcije
6:                  expression formulacija1
```

Formalizem bomo demonstrirali na enostavnem primeru konceptualnega modela (Slika 2). Kot smo že povedali, znanje v knjižnici kodiramo v obliki generičnih procesov. Torej shema predstavlja neko generalizirano znanje, ki ga lahko apliciramo na sisteme z istimi tipi spremenljivk. S tem znanjem v knjižnici lahko zgeneriramo (torej specializiramo) tako enostavni model s slike 1, kot tudi malenkost kompleksnejši za dve hranili in eno vrsto fitoplanktona (Slika 3), kakor tudi še bolj kompleksen model za npr. dve anorganski hranili in tri različne vrste fitopolanktona.

2.3.1 Taksonomija spremenljivk

V sistemu imamo dva tipa spremenljivk, t.j. anorgansko hranilo in primarni producent. Oba tipa sta izražena s koncentracijo (masa/volumen). Zato lahko deklariramo tip *Concentration* in dva podtipa: (1) *Inorganic*, ki predstavlja koncentracijo anorganskih hranil in (2) *Primary producer*, ki predstavlja koncentracijo primarnih producentov. V primeru, da želimo modelirati interakcijo med več vrstami (npr. primarni producent, ki se hrani z dvemi hranili) deklariramo množico določenega tipa spremenljivk, (Tabela 3, vrstice 9, 10 in 11). Vrstica, ki se začne z znakom #, predstavlja komentar.

Tabela 3: Taksonomija tipov spremenljivk v domenski knjižnici

```
 1:  # Deklaracija generičnih tipov spremenljivk
 2:          type Concentration is real
 3:
 4:  # Deklaracija generičnih (pod)tipov, generične spremenljivke Concentration
 5:          type Inorganic is Concentration
 6:          type Primary_producer is Concentration
 7:
 8:  # Deklaracija množic posameznih (pod)tipov
 9:          type Concentrations is set (Concentration)
10:          type Inorganics is set(Inorganic)
11:          type Primary_producers is set(Primary_producer)
```

2.3.2 Taksonomija procesnih razredov v sistemu

Procesni razred predstavlja končno množico poznanih matematičnih formulacij (modelov) procesa v sistemu. V našem sistemu imamo dva procesna razreda. Prvi predstavlja rast primarnega producenta, drugi pa respiracijo. Glede na domensko znanje predstavljeno v oddelku 2.1, vsebuje procesni razred *rast* dva modela, t.j. eksponentna rast (3) ali omejena rast (4). Model za omejeno rast bomo zaradi enostavnosti v nadaljevanju upoštevali brez temperaturnega vpliva, torej brez funkcije f1(T). Rast primarnega producenta bo torej omejena samo s koncentracijami hranil. Ta dva modela predstavljata podrazreda procesnega razreda *rast*. Deklaracijo procesnih razredov kaže Tabela 4. Deklaracija procesnega razreda *rast* je prikazana v vrsticah od 1 do 9. Ime procesnega razreda je *Growth_PP* in v svoji deklaraciji vsebuje dva tipa spremenljivk, to so *Primary producer* (*pp*), ki predstavlja primarnega producenta in *Inorganics* (*cs*), ki predstavlja množico anorganskih hranil, s katerimi se hrani primarni producent. Če bi spremenljivka v procesu imela podtipe, bi proces deloval tudi na te podtipe. Procesni razred vsebuje dva podrazreda, oz. dve različni formulaciji procesa *rast*. Prvi je eksponentna rast, ki je definiran v vrsticah 5 in 6. Ime podrazreda je *Exponential_growth*, njegova formulacija pa je zapisana v vrstici 6. Drugi podrazred je omejena rast. Ime podrazreda je *Limited_growth* (vrstica 8), formulacija pa je prikazana v vrstici 9. Opazimo, da v tej formulaciji nastopa člen: **product({c}, c in cs, c/(const(saturation, 0, 1, 2) + c)**. Produkt multiplikativno kombinira omejitvene funkcije posameznega hranila za rast fitoplanktona (7), t.j. enakovreden je matematičnemu izrazu $\prod_i \frac{c_i}{const + c_i}$. Pogoj *c in cs* pomeni, da v produktu nastopajo le tista hranila ki se nahajajo v množici hranil c_s. Npr. za i= 2 bomo imeli:

$\frac{c_1}{const(saturation,0,1,2) + c_1} \cdot \frac{c_2}{const(saturation,0,1,2) + c_2}$, kjer sta c_1 in c_2 elementa množice c_s.

Dugi procesni razred *respiracija* je deklariran v vrsticah od 12 do 15. Ime razreda je *Respiration_PP*. Ima samo eno možno formulacijo, torej en podrazred (vrstici 14 in 15).

Uporaba funkcijskih razredov

Omenili smo, da formalizem podpira tudi deklaracijo funkcijskih razredov. Funkcijski razredi so uporabni, ko želimo vpeljati nek izraz (vpliv) v določenem procesnem razredu, ki pa ima lahko več možnih formulacij. Za prikaz uporabnosti funkcijskih razredov bomo naše domensko znanje nekoliko razširili. Trenutno naše znanje vsebuje eno samo formulacijo omejitvene funkcije anorganskega hranila za rast primarnega producenta. Formulirali smo jo z Monodovim modelom (6) in (7). Znano je, da to funkcijo lahko formuliramo z več različnimi modeli. Recimo, da naše domensko znanje o omejitvenih funkcijah razširimo še z funkcijo Monod[2] (13):

$$f(nut) = \frac{nut^2}{k5 + nut^2} \qquad (13)$$

Tako lahko sedaj funkcija *f2* v en. 4 zavzame dve formulaciji – Monod ali Monod^2.

Tabela 4: Taksonomija procesnih razredov v domenski knjižnici

1:	# Deklaracija prvega generičnega procesnega razreda rast primarnih producentov:
2:	**process class** Growth_PP(Primary_producer pp, Inorganics cs)
3:	
4:	#Deklaracija (pod)razredov, ki predstavljajo formulacije razreda Growth_PP
5:	**process class** Exponential_growth **is** Growth_PP
6:	**expression const**(gr_rate, 0, 0.5, 2) * *pp*
7:	# **const**(gr_rate, 0, 0.5, 2) predstavlja konstanto, t.j. parameter hitrosti rasti (gr_rate) s spodnjo mejo 0, začetno vrednostjo 0.5 in zgornjo mejo 2
8:	**process class** Limited_growth **is** Growth_PP
9:	**expression const**(gr_rate, 0, 0.5, 2) * pp * product({c}, c in cs, c/(const(saturation, 0, 1, 2) + c)
10:	
11:	# Deklaracija generičnega procesnega razreda respiracija primarnih producentov:
12:	**process class** Respiration_PP(Primary_producer pp)
13:	# Deklaracija (pod)razreda
14:	**process class** Exponential_resp **is** Respiration_PP
15:	**expression const**(resp_rate, 0, 0.5, 2) * *pp*
16:	# **const**(resp_rate, 0, 0.5, 2) predstavlja konstanto, t.j. parameter hitrosti odmiranja (respiracije: resp_rate) s spodnjo mejo 0, začetno vrednostjo 0.5 in zgornjo mejo 2

V knjižnici smo omejitveno funkcijo upoštevali v drugem podrazredu procesnega razreda *Growth_PP* (Tabela 4, vrstica 8) tako, da smo jo formulirali z Monodovim modelom ((6). Zgoraj opisano razširitev domenskega znanja lahko v knjižnici upoštevamo z deklaracijo dodatnega podrazreda procesnega razreda *Growth_PP*. Sedaj imamo dve možni formulaciji (dva podrazreda) za omejitveni model (Tabela 5), *Limited_growth1* (vrstica 8), ki upošteva formulacijo omejitvene funkcije hranila z Monodovim modelom in *Limited_growth2* (vrstica 11), ki to upošteva z modelom Monod^2.

Taka deklaracija procesnega razreda je ustrezna, če želimo formulirati model z enim samim anorganskim hranilom. V tem primeru bo proces Growth_PP pravilno formuliran po enem izmed treh deklariranih podrazredov, t.j. exponentna rast, omejitvena z uporabo Monodove funkcije ali omejitvena z uporabo funkcije Monod^2. V primeru več omejitvenih hranil pa ti trije podrazredi ne pokrivajo vseh možnih formulacij procesa Growth_PP. Če imamo dve hranili (c_1 in c_2) bo model omejitvene rasti formuliran bodisi po (14) ali pa po (15).

$$\text{omejitvena\_rast} = \text{const\_gr} \cdot \frac{c_1}{\text{const\_sat} + c_1} \cdot \frac{c_2}{\text{const\_sat} + c_2} \cdot pp \qquad (14)$$

$$\text{omejitvena\_rast} = \text{const\_gr} \cdot \frac{c_1^{\,2}}{\text{const\_sat} + c_1^{\,2}} \cdot \frac{c_2^{\,2}}{\text{const\_sat} + c_2^{\,2}} \cdot pp \tag{15}$$

Ne bo pa možna kombinacija omejitvenih funkcij v modelu, kot je recimo formulacija, ki jo kaže enačba (16).

$$\text{omejitvena\_rast} = \text{const\_gr} \cdot \frac{c_1}{\text{const\_sat} + c_1} \cdot \frac{c_2^{\,2}}{\text{const\_sat} + c_2^{\,2}} \cdot pp \tag{16}$$

Tabela 5: Vpeljava dodatnega podrazreda v procesni razred Growth_PP

```
1:  # Deklaracija procesnega razreda rast primarnih producentov, s tremi
    podrazredi:
2:      process class Growth_PP(Primary_producer pp, Inorganics cs)
3:
4:      #Deklaracija (pod)razredov, ki predstavljajo formulacije razreda
    Growth_PP
5:      process class Exponential_growth is Growth_PP
6:          expression const(gr_rate, 0, 0.5, 2) * pp
8:      process class Limited_growth1 is Growth_PP
9:          expression const(gr_rate, 0, 0.5, 2) * pp * product({c}, c in cs,
                        c/(const(saturation, 0, 1, 2) + c)
10:     process class Limited_growth2 is Growth_PP
11:         expression const(gr_rate, 0, 0.5, 2) * pp * product({c}, c in cs,
                        c*c/(const(saturation, 0, 1, 2) + c*c)
```

Zato je smiselna vpeljava funkcijskega razreda, ki vsebuje ti dve funkciji. Funkcijski razred *Food_limitation*, ki vsebuje formulacije, ki jih lahko uporabimo kot omejitvene funkcije za rast primarnih producentov prikazuje Tabela 6.

Tabela 6: Deklaracija funkcijskega razreda Food_limitation

```
1:  # Deklaracija funkcijskega razreda omejitvena funkcija, z dvema
    podrazredoma:
2:      function class Food_limitation(Inorganic c)
3:
4:      #Deklaracija (pod)razredov - formulacije razreda Food_limitation
5:      function class Food_limitation_type_1() is Food_limitation
6:          expression c / (c + const(saturation_rate, 0, 0.02, 10))
8:      function class Food_limitation_type_2() is Food_limitation
9:          expression c * c / (c * c + const(saturation_rate, 0, 0.02,10))
```

Ustrezno vkomponirana funkcija Food_limitation v procesni razred Growth _PP omogoča generiranje vseh možnih modelov za različno število omejitvenih hranil.

Vpeljavo funkcije v procesni razred Growth_PP kaže Tabela 7, vrstica 9. Taka definicija omejitvenega modela je v skladu z enačbo (17):

$$rast = \mu \cdot pp \cdot \text{Food\_limitation(nut)} \tag{17}$$

in pri tem upošteva vse možne kombinacije omejitvene funkcije (Food_limitation) anorganskih hranil. Uporaba knjižnice, oz. generiranje modelov za specifične opazovane sisteme iz knjižnice je opisano v poglavju 1.4.

Tabela 7: Končna deklaracija procesnega razreda Growth_PP

```
1:  # Deklaracija procesnega razreda rast primarnih producentov:
2:      process class Growth_PP(Primary_producer pp, Inorganics cs)
3:
4:      #Deklaracija (pod)razredov, ki predstavljajo formulacije razreda
    Growth_PP
5:      process class Exponential_growth is Growth_PP
6:          expression const(gr_rate, 0, 0.5, 2) * pp
8:      process class Limited_growth is Growth_PP
9:          expression const(growth_rate, 0, 0.5, 2)*pp*product({c},c in
    cs,                 Food_limitation(c))
```

2.3.3 Kombinatorne sheme

Kombinatorne sheme predstavljajo masne bilance posameznih tipov sistemskih (odvisnih) spremenljivk. Z njimi ustrezno kombiniramo procesne razrede v model celotnega sistema. V naši enostavni domeni imamo dva tipa odvisnih spremenljivk (*Inorganic* in *Primary_producer*), torej potrebujemo dve kombinatorni shemi za model celotnega sistema (Tabela 8). Časovni odvod za posamezen tip sistemske spremenljivke zapišemo z rezervirano funkcijo **time_deriv**(ime spremenljivke) (vrstici 3, 9).

Tabela 8: Kombinatorne sheme posameznih tipov sistemskih (odvisnih) spremenljivk v domenski knjižnici

```
1:  #Kombinatorna shema (masna bilanca) anorganskega hranila
2:  combining scheme Lake(Inorganic i)
3:      time_deriv(pp) =
4:          - sum({pp}, true, const(conv_fact,0,0.01,1)*Growth_PP(pp, i))
5:          +          sum({pp},          true,
6:  const(conv_fact,0,0.01,1)*Respiration_PP(pp))
7:
8:  #Kombinatorna shema (masna bilanca) primarnega producenta
9:  combining scheme Lake(Primary_producer pp)
10:     time_deriv(pp) =
11:         + sum({food}, true, Growth_PP(pp, food))
            - sum({}, true, Respiration_PP(pp))
```

Opazimo, da vse procesne razrede kombiniramo (seštevamo) preko funkcije *sum*. Funkcija je sestavljena iz treh delov. V prvem delu se v zavitem oklepaju nahaja tip spremenljivke po katerem seštevamo. Drugi del vsebuje pogoj, ki se upošteva pri seštevanju, npr. pogoj *true* pomeni, da se sešteva po vseh spremenljivkah pp, tretji pa predstavlja ustrezno zapisan procesni razred. Funkcija v prvem členu prve kombinatorne sheme sešteje vse izraze procesa Growth_PP (sešteva se po prvem argumentu v procesnem razredu), v katerih naključni primarni producent {pp} konzumira hranilo i. Če imamo npr. dva primarna producenta (pp1 in pp2) v opazovanem sistemu, bomo imeli seštevek dveh procesov: Growth_PP(pp1,i) in Growth(pp2,i). V drugem členu funkcija *sum* sešteje vse prispevke zaradi respiracije naključnega pp k hranilu i. Uporaba funkcije se zdi nepotrebna v drugi kombinatorni shemi, saj se procesa Growth_PP in Respiration tu nanašata le na primarni producent pp, oz. prvi proces pomeni rast primarnega producenta pp, drugi pa respiracijo primarnega producenta pp. V takih primerih je uporaba funkcije ugodna, ko se določen proces ne pojavlja v opazovanem sistemu. Vrednost člena v zgeneriranih modelnih strukturah bo v tem primeru enaka nič, torej brez vpliva na masno bilanco. Naslednje poglavje natančneje opisuje uporabo knjižnice in njeno transformacijo glede na specifikacijo opazovanega sistema v modelne strukture.

2.4 Uporaba domenske knjižnice za generiranje modelov - specifikacija opazovanega sistema

V prejšnjem poglavju (1.3) smo opisali enostavno domensko knjižnico, ki vsebuje generalizirano znanje o problemski domeni, v našem primeru prehranjevalni mreži v jezeru. V specifikaciji opazovanega sistema ekspert (uporabnik) poda svoje znanje o določenem (specifičnem) sistemu, ki ga želi modelirati. Na podlagi te specifikacije LAGRAMGE 2.0 določi prostor ustreznih struktur modela. Z drugimi besedami tu podajamo konceptualni model opazovanega sistema. Specifikacija vključuje deklaracijo spremenljivk in procesov, ki nastopajo v sistemu. Spremenljivke deklariramo na naslednji način:

variable tip_spremenljivke ‘ime_spremenljivke’

Besedo **system** postavimo pred besedo **variable** če je spremenljivka sistemska., oz. če želimo, da LAGRAMGE odkrije enačbo za to spremenljivko. Zunanje, oz. gonilne spremenljivke so privzete kot poznane (merjene) in za njih LAGRAMGE 2.0 ne išče opisne enačbe.

Proces v sistemu definiramo z besedo *process*, imenom procesa, kot je to deklarirano v knjižnici in ustreznimi argumenti t.j. spremenljivkami ustreznega tipa, ki nastopajo v tem procesu:

process ime_procesa (*argument1, argument2...*) **oznaka_procesa**

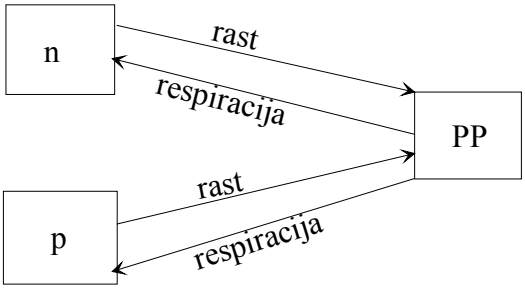
Razvidno je, da za pravilno zapisano specifikacijo opazovanega sistema potrebujemo znanje o tipih spremenljivk in procesnih razredih deklariranih v knjižnici. V našem primeru knjižnice smo deklarirali tri tipe spremenljivk

(Concentration, Inorganic in Primary_producer). Torej lahko v opazovanem sistemu nastopajo samo spremenljivke teh tipov. Tabela 9 podaja povzetek deklaracije procesnih razredov, kjer so zbrani potrebni podatki za pravilno specifikacijo procesov. V prvem stolpcu je podan opis procesnega razreda. Drugi stolpec vsebuje imena procesnih razredov, medtem ko so v tretjem in četrtem stolpcu podatki o argumentih, oz. tipih spremenljivk, ki nastopajo v procesnem razredu. Iz tretjega stolpca lahko razberemo, koliko argumentov vsebuje določen proces in katerega tipa so, iz četrtega pa, ali je določen argument definiran kot množica.

Tabela 9: Povzetek deklaracij procesnih razredov v knjižnici

	Opis procesnega razreda	Ime procesnega razreda	Argumenti, oz. tipi spremenljivk, ki nastopajo v procesu	Argument deklariran kot množica: da/ne
1	Rast primarnega producenta	Growth_PP	Primary_producer Inorganic	ne da
2	Respiracija primarnega producenta	Respiration	1. Primary producer	ne

Specifikacijo sistema bomo prikazali na enem izmed primerov, ki ga lahko modeliramo s to enostavno knjižnico. Konceptualni model opazovanega sistema prikazuje Slika 4. Primer opisuje dinamiko fitoplanktona (PP) in dveh anorganskih hranil (n in p). Koncentracija fitoplanktona narašča zaradi porabe hranil, upada pa zaradi respiracije. Ravno nasprotno se dogaja s hranili n in p.



Slika 4: Konceptualni model dinamike primarnega producenta in dveh anorganskih hranil

Specifikacijo sistema kaže Tabela 10. V našem primeru imamo dve spremenljivki tipa *Inorganic* (n in p) in eno spremenljivko tipa *Primary_producer* (pp). Deklaracija opazovanih spremenljivk je podana v vrsticah 2 do 4, medtem ko sta oba procesa, t.j. rast in respiracija primarnega producenta deklarirana v zadnjih dveh vrsticah (7 in 8). Za pravilno specifikacijo procesov si pomagamo s Tabela 9. Ime procesa *rast primarnega producenta* je *Growth_PP*. Prvi argument je tipa Primary_producer (v našem primeru je to spremenljivka phyto), drugi pa predstavlja množico anorganskih hranil, ki jih primarni producent konzumira. Množico zapišemo v zavitih oklepajih {}. Glede na konceptualni model (Slika 4), množica vsebuje obe anorganski hranili (vrstica 7). Ime drugega procesa (respiracija

primarnega producenta) je *Respiration*. Vsebuje en argument tipa Primary_producer (phyto).

Tabela 10: Specifikacija opazovanega sistema iz Slika 4

1:	#Specifikacija spremenljivk v opazovanem sistemu (Slika 4)
2:	**variable** Inorganic *n*
3:	**variable** Inorganic *p*
4:	**variable** Primary producer *phyto*
5:	
6:	#Specifikacija procesov v opazovanem sistemu ((Slika 4).
7:	**process** Growth_PP (*phyto*,{*n*, *p*}) gr
8:	**process** Respiration (*phyto*) resp

2.4.1 Pretvorba specifikacije sistema v gramatiko

Podana specifikacija se nadalje avtomatsko transformira v gramatiko, oz. prostor možnih struktur modelov za podani sistem. Najprej se zapišejo masne bilance sistemskih spremenljivk. Za naš primer dobimo naslednje časovne odvode odvisnih spremenljivk:

$$n' = -\text{const}(conv\_fact,0,0.01,1)\cdot \text{Growth}(phyto,\{n,p\}) + \text{const}(conv\_fact,0,0.01,1)\cdot \text{Respiration}(phyto)$$
$$p' = -\text{const}(conv\_fact,0,0.01,1)\cdot \text{Growth}(phyto,\{n,p\}) + \text{const}(conv\_fact,0,0.01,1)\cdot \text{Respiration}(phyto)$$
$$phyto' = +\text{Growth}(phyto,\{n,p\}) - \text{Respiration}(phyto)$$

V naslednjem koraku se procesni razredi v masnih bilancah transformirajo v ustrezne modele. Procesni razred *Growth_PP* se transformira v dve modelni strukturi, kot je prikazano v Tabela 11. Druga stuktura procesa (vrstica 2) vsebuje funkcijski razred *Food_limitation(n,p)*, ki predstavlja omejitveno funkcijo hranil na rast fitoplanktona. Ker obravnavamo dve omejitveni hranili, je skupni vpliv obeh hranil enak produktu funkcij posameznega hranila (vrstica 3), kot smo definirali v domenski knjižnici. Limitirajoča funkcija vsakega hranila ima po dve možni formulaciji (vrstice od 4 do 11). Torej obstajajo štiri formulacije za funkcijski razred *Food_limitation(n, p)* in skupaj pet formulacij za procesni razred Growth_PP (Tabela 11). Ker obstaja za procesni razred *Respiration* le en model v domenski knjižnici, ostaja število možnih modelov za zadano specifikacijo pet. Strukture modelov kaže Tabela 12. Koeficienti k1 do k8 so parameteri modelov.

Tabela 11: Transformacija procesnih razredov Growth_PP in Respiration iz specifikacije sistema (Tabela 10) v gramatiko modelov, kot jo avtomatsko izvede LAGRAMGE 2.0 v prvi stopnji

```
1:      Growth(phyto,{n,p})= const*phyto
2:      Growth(phyto,{n,p}) = const*phyto*Food_limitation(n,p)
3:
4:      Food_limitation(n,p) = Food_limitation(n)* Food_limitation(p)
5:      Food_limitation(n) = Food_limitation_type_1(n)
6:      Food_limitation(n) = Food_limitation_type_2(n)
7:
8:      Food_limitation_type_1(n) = n/(const + n)
9:      Food_limitation_type_2(n) = n*n/(const + n*n)
10:
11:     Food_limitation(p) = Food_limitation_type_1(p)
12:     Food_limitation(p) = Food_limitation_type_2(p)
13:
14:     Food_limitation_type_1(p) = p/(const + p)
15:     Food_limitation_type_2(p) = p*p/(const +p*p)
16:
17:     Respiration (phyto)= const(resp_rate,0,0.5,2)*phyto
```

2.4.2 Optimizacija dobljenih struktur modelov

Druga stopnja v postopku avtomatske indukcije modelov z uporabo predznanja z orodjem LAGRAMGE 2.0 je optimizacija vseh zgeneriranih modelov. Za ta korak potrebujemo časovne meritve spremenljivk v sistemu, na katere LAGRAMGE 2.0 umerja konstantne parametre.

Tabela 12: Gramatika oz. prostor možnih modelov za specifikacijo sistema kot kaže Tabela 10

Struktura 1 $n' = -k1 \cdot k2 \cdot phyto + k3 \cdot k4 \cdot phyto$ $p' = -k5 \cdot k2 \cdot phyto + k6 \cdot k4 \cdot phyto$ $phyto' = k2 \cdot phyto - k4 \cdot phyto$	Struktura 2 $n' = -k1 \cdot k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p}{k8+p} + k3 \cdot k4 \cdot phyto$ $p' = -k5 \cdot k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p}{k8+p} + k6 \cdot k4 \cdot phyt$ $phyto' = k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p}{k8+p} - k4 \cdot phyto$
Struktura 3 $n' = -k1 \cdot k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p}{k8+p} + k3 \cdot k4 \cdot phyt$ $p' = -k5 \cdot k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p}{k8+p} + k6 \cdot k4 \cdot phyt$ $phyto' = k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p}{k8+p} - k4 \cdot phyto$	Struktura 4 $n' = -k1 \cdot k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p^2}{k8+p^2} + k3 \cdot k4 \cdot phy$ $p' = -k5 \cdot k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p^2}{k8+p^2} + k6 \cdot k4 \cdot phy$ $phyto' = k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p^2}{k8+p^2} - k4 \cdot phyto$
Struktura 5 $n' = -k1 \cdot k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p^2}{k8+p^2} + k3 \cdot k4 \cdot phy$ $p' = -k5 \cdot k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p^2}{k8+p^2} + k6 \cdot k4 \cdot phy$ $phyto' = k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p^2}{k8+p^2} - k4 \cdot phyto$	

3 Modeliranje procesov v vodnem ekosistemu

3.1 Struktura in funkcija vodnih ekosistemov

Za razumevanje vodnega ekosistema je potrebno poznavanje njegove strukture in funkcije (Overbeck, 1989). Strukturo sistema določajo abiotske in biotske komponente. Med abiotske komponente prištevamo:

- *Anorganske snovi*. Sem spadajo ioni, t.i hranila in esencialni elementi v zelo majhnih koncentracijah (Fe, Mn, Zn, Cu, Mo, Si, itd.). Ionska sestava površinske vode je predvsem odvisna od preperevanja kamnin, padavin in razmerja med padavinami in evaporacijo. Med kationi prevladujejo Ca^{2+}, Mg^{2+}, Na^{+}, K^{+}, med anioni pa HCO_3^{-}, CO_3^{2-}, SO_4^{2-}. Hranila so tiste anorganske snovi, ki jih asimilirajo primarni producenti. Med hranila, ki v večini primerov najbolj vplivajo na primarno produkcijo v jezerih in rekah, prištevamo dušik in fosfor. Glede na to, da je fosforja v naravi relativno malo (manj kot dušika glede na sestavo organske snovi v živih organizmih), le ta običajno omejuje primarno produkcijo.
- *Organska snov*. Glavne organske komponente so ogljikovi hidrati, proteini, pigmenti, vitamini. Nastajajo v metabolnih procesih v celicah in so zelo pomembni v vodnem ekosistemu kot ekstracelularna raztopljena organska snov.
- *Klimatski pogoji*, kot npr. temperatura, svetloba in veter so izrednega pomena pri delovanju ekosistema.

Biotske komponente vodnega ekosistema so (1) producenti organske snovi (avtotrofni organizmi), (2) makropotrošniki (zooplankton in ribe), (3) mikropotrošniki (heterotrofne bakterije, ki razkrajajo raztopljeno in suspendirano organsko snov, ki jo producirajo avtotrofi). Biomaso lahko v splošnem razdelimo na dve komponenti – avtotrofna in heterotrofna, ki sta med sabo povezani preko metabolnih procesov v prehranjevalni verigi.

Funkcija ekosistema je določena z njegovo dinamiko oz. hitrostjo transformacij posameznih komponent iz ene oblike v drugo (npr. iz anorganske v organsko). Dinamiko lahko analiziramo preko (1) kroženja energije, (2) kroženja hranil, (3) prehranjevalne verige in (4) omejitve in kontrole metabolnih procesov.

Torej, ekosistem kot funkcionalna enota vključuje biotsko in abiotsko okolje, ki sta med seboj povezani in vplivata (omejujeta) eno na drugo (Overbeck, 1989).

3.2 Izmenjava in transformacije snovi v vodnem ekosistemu

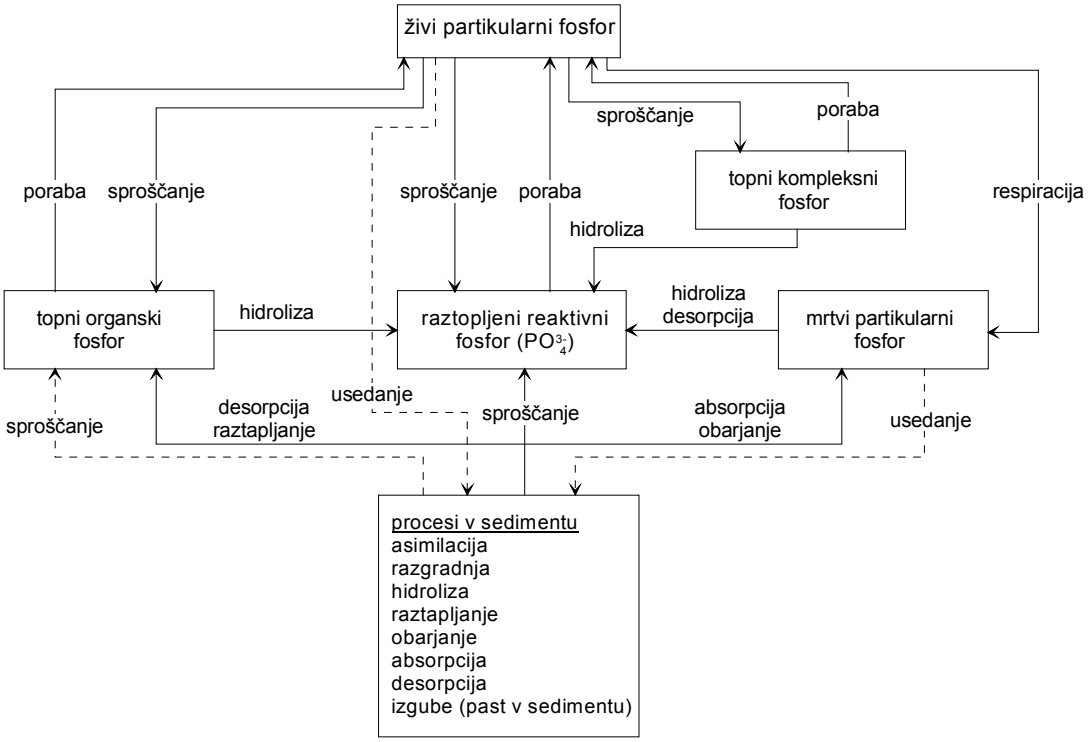
V vodnem ekosistemu nenehno potekajo številni procesi, preko katerih se anorganska snov transformira v organsko snov in obratno, organska snov se transformira v anorgansko s pomočjo mikroorganizmov.

Anorganska hranila dosežejo vodni sistem bodisi preko zunanje in/ali notranje obremenitve. Zunanja obremenitev pravimo hranilom, ki pridejo v sistem od zunaj, t.j. s padavinami, z izpiranjem iz prispevne površine ali s pritoki. Hranila, ki se

sproščajo v vodni sistem iz samega sistema preko izločkov vseh živih organizmov, z mineralizacijo mrtve organske mase, s sproščanjem iz sedimenta ter preko hidrolize raztopljene organske mase, predstavljajo notranjo obremenitev sistema. Raztopljena anorganska hranila konzumirajo fitoplankton in druge vodne rastline med fotosintezo, ki se preko prehranjevalne mreže razširijo v druge organizme (rastlinojedi in mesojedi). Suspendirani delci organske mase pridejo v vodo preko suspendiranih izkočkov vodnih živali in odmiranja planktonskih organizmov (detritus), ki jih bakterije v svojem metabolnem procesu vrnejo nazaj v sistem kot anorgansko snov. Mrtva organska snov in fitoplankton prideta v sediment z usedanjem. Z razkrojem suspendirane mrtve organske snovi in organskega sedimenta pride do sproščanja raztopljenih organskih in anorganskih snovi.

3.2.1 Fosforjev krog

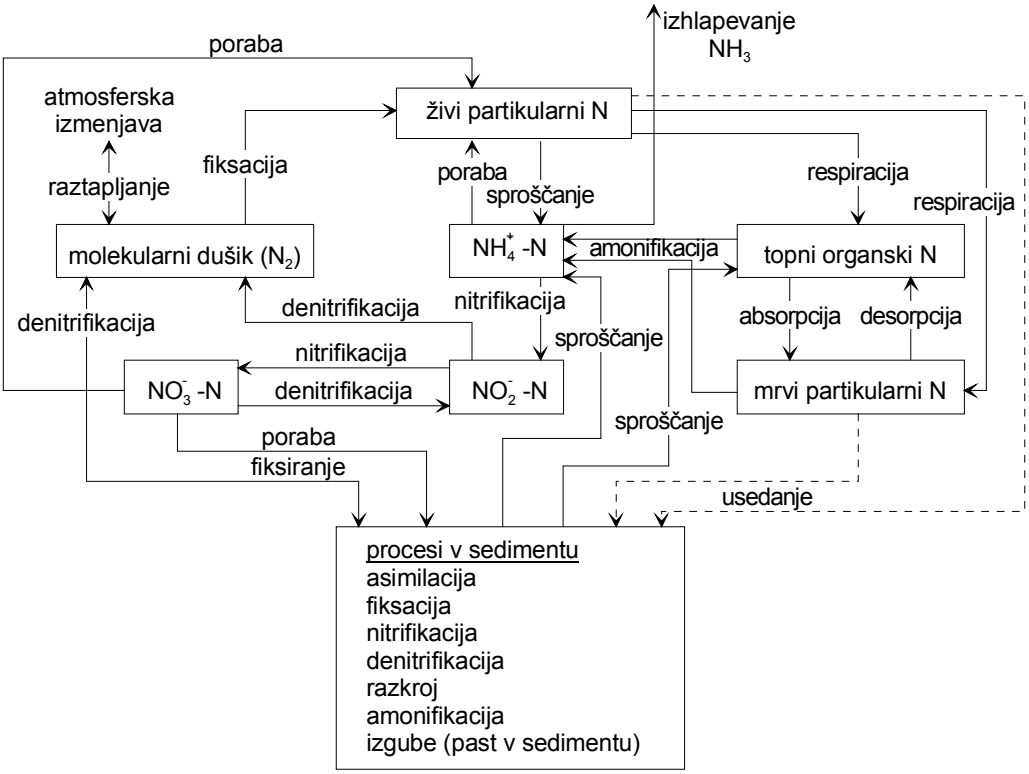
Fosfor se v vodnih ekosistemih nahaja kot partikularni fosfor ter kot raztopljeni organski in anorganski fosfor. Partikularni fosfor predstavlja fosfor vezan v mikroorganizmih, algah, drugih rastlinah in živalih, fosfor adsorbiran na anorganske komponente kot so gline, karbonati in železovi hidroksidi ter fosfor adsorbiran na mrtvo partikularno organsko materijo. Raztopljeni fosfor v vodi sestoji iz ortofosfatov (PO_4^{3-}), polifosfatov, ki so v glavnem sintetičnega izvora (detergenti, kemični stabilizatorji vode v ogrevalnih sistemih), in organskega fosforja (izločki organizmov). Kroženje fosforja v vodnem ekosistemu prikazuje Slika 5



Slika 5: Kroženje fosforja v vodnih sistemih (Bowie et al., 1985)

3.2.2 Dušikov krog

V vodah se dušik pojavlja kot: vezan organski dušik (v proteinih, amino kislinah, sečnini, itd.), amonij (NH_4^+), amoniak (NH_3), nitritni dušik (NO_2^{2-}) ali nitratni dušik (NO_3^-). Elementarni dušik pretvarjajo mikroorganizmi v amonijak (fiksiranje dušika). Rastline asimilirajo dušik v obliki amonijaka in nitrata ter ga vgrajujejo v beljakovine. Nekateri primarni producenti, kot so modrozelene alge (oziroma bakterije), imajo sposobnost asimilacije tudi elementarnega dušika iz ozračja. Iz rastlinskih prehaja dušik v živalske beljakovine s prehrano. Končni produkt pri večini živali je sečnina, ki se s pomočjo mikroorganizmov pretvori v amonijak, nitrite in nitrate. Kroženje dušika v vodnih sistemih prikazuje Slika 6.



Slika 6: Kroženje dušika v vodnih sistemih (Bowie et al., 1985)

3.3 Sloji v vodnem telesu

Sloji oz. cone v vodnem telesu so določeni z ozirom na prevladujoči biokemijski proces in gostoto vode. Glede na prevladujoči biokemijski proces se vodno telo oz limnetična ali pelagična cona razdeli na trofogeno in trofolitsko cono. Trofogena cona je zgornji, dobro premešan in presvetljen sloj, kjer prevladuje proces primarne produkcije. Pravimo ji tudi epilimnij, čeprav to ni identično. Če je koncentracija primarnih producentov (fitoplanktona) visoka, potem je tudi absorbcija svetlobe velika, kar pomeni, da svetloba sega največ do spodnje meje epilimnija. V obratnem primeru pa lahko svetloba doseže tudi zgornje sloje hipolimnija in tudi tam omogoča fotosintezo. Globini, kjer sta fotosinteza in respiracija izenačeni, rečemo kompenzacijska globina. V trofolitski coni prevladujeta procesa respiracije in dekompozicije. Mejo med trofogeno in trofolitsko cono določa predvsem globina,

do katere sega svetloba. Bentična cona predstavlja dno jezera in je razdeljena na litoralno in profundalno cono. Litoralna cona sega od meje med jezerom in kopnom do globine, kjer so še koreninsko pritrjene vodne rastline. Nadalje je področje profundalne cone. V bentični coni poteka velika aktivnost heterotrofnih bakterij, kar pomeni, da je transformacija organske (raztopljene in partikularne) snovi razmeroma hitra.

Gostotno slojevanje vode je značilno za globoke sisteme. Ker svetloba ne doseže spodnjih slojev, se segreje le zgornji sloj - epilimnij, ki postane lažji. Ta sloj je dobro premešan in bogat s kisikom zaradi primarne produkcije. Hipolimnij je spodnji, hladnejši in gostejši sloj, ki se slabo meša z epilimnijem. Med hipolimnijem in epilimnijem se nahaja termoklina, t.j. vmesni sloj, kjer temperatura zelo hitro upada z globino. Termoklina je omejena na vmesno cono – metalimnij - kjer temperatura pada za vsaj 1°C na en meter. V plitvih sistemih sega svetloba do dna, kar običajno pomeni, da ni slojitve. Primarna produkcija poteka po celotnem vodnem stolpcu, temperatura pa je bolj ali manj konstantna po celi globini (Overbeck, 1989).

3.4 Jezerski in rečni ter morski obalni ekosistem

Večina ekoloških procesov, ki se odvijajo v jezerskih ekosistemih, deluje tako v obalnih (lagune zalivi, ustja rek, fjordi,…) kot v drugih vodnih ekosistemih. Vendar pa ima vsak izmed teh ekosistemov nekoliko drugačne robne pogoje, ki jih je treba upoštevati pri določanju funkcije sistema.

Glavna razlika med jezerskimi in rečnimi ter morskimi ekosistemi je v izmenjavi vode, ki je v slednjih bistveno večja, zaradi vzdolžnega transporta (reke, morski tokovi) in vertikalnega ter horizontalnega delovanja bibavice. S tem je povezan transport (v in iz sistema) ekološko pomembnih snovi. Druga pomembna razlika je slanost vode. Voda v morskih obalnih ekosistemih je ponavadi mešanica sladke in slane vode. Organizmi v takem sistemu razvijejo posebne mehanizme, s katerimi izločajo sol iz svojih tekočin. Tem mehanizmom je v pomoč tudi sposobnost sintetiziranja organskih snovi, med katerimi je tudi dimetilsulfoniopropionat (DMSP), ki vzdržujejo osmotsko ravnotežje v tekočinah. DMSP razpada v dimetilsulfid (DMS), kar pomeni, da so lahko ti sistemi glavni vir žvepla v atmosferi. Poleg DMS imajo morski obalni ekosistemi tudi sicer večjo koncentracijo sulfata kot jezerski. Razlika v slanosti vodnih mas povzroča dodatno razslojevanje (poleg temperaturnega) v vodnem stolpcu, kar organizmom omejuje dostopnost potrebnih elementov. Razliko med jezerskim in morskim obalnim ekosistemom je mogoče opaziti tudi v kroženju hranil. Hranila prihajajo v sistem (1) preko padavin, z izpiranjem iz prispevne površine in v primeru dušika s fiksacijo atmosferskega dušika in (2) s sproščanjem hranil preko razgradnje raztopljene in mrtve organske mase in z izločki živih organizmov – notranja obremenitev. Medtem ko je v jezerskih ekosistemih (predvsem globjih) večinoma fosfor omejitveno hranilo za primarno produkcijo, saj je dušika dovolj v atmosferi, je za obalne ekosisteme značilno (potrjeno z meritvami), da običajno dušik omejuje primarno produkcijo. Možne razloge lahko iščemo v količini fiksiranega dušika. Fiksacija dušika je

občutno večja v sladkovodnh kot v obalnih sistemih. Za to obstajata vsaj dve hipotezi (Valiela, 1991):

- Visoka koncentracija sulfata lahko ovira fiksacijo dušika. Sulfatni ion je zelo podoben molibdenu, ki je ključni element encimskega sistema za fiksacijo dušika. Tako pri veliki koncentraciji sulfata lahko celice namesto molibdena 'pomotoma' asimilirajo sulfat in je fiksacija dušika zato reducirana.
- Skupki organskih snovi ali celic formirajo majhne anaerobne cone, ki so potrebne pri fiksaciji dušika. Turbulenca, ki je bolj izrazita v obalnih ekosistemih, pa uničuje te skupke in s tem preprečuje fiksacijo.

Na tem mestu velja opozoriti, da so tudi nekatera jezera omejena z dušikom. Verjetno je vzrok predvsem v tem, kakšen je zadrževalni čas hipolimnijske vode (saj je v epilimniju dušik na razpolago fiksatorjem dušika). Torej plitva jezera in plitve lagune, ki imajo bogate sedimente in pritjeno biomaso v njih, ne občutijo pomanjkanja fosforja kljub oksičnim razmeram – torej so omejene z dušikom, zaradi počasnega pretoka elementarnega dušika iz atmosfere. To trditev smo delno potrdili z nekaterimi aplikativnimi modeli razvitimi v okviru naloge (poglavje 5, jezero Kasumigaura).

Denitrifikacija lahko odstrani bistveno več dušika v obalnih sistemih, kot v jezerskih. Čeprav je denitrifikacija odvisna od obremenitve – nižja pri visokih obremenitvah – se pri morskih ekosistemih odstrani vsaj 40-50 % prihajajočega dušika (ne glede na obremenitev) medtem ko pri jezerskih sistemih 10-35 % dušika (Valiela, 1991). Obalni sistemi so ponavadi plitvejši (svetloba do dna in prim prod. po celem stolpcu). Če imamo anoksični sediment, se fosfat sprošča (iz železovega fosfata) in ker ni stratifikacije kontinuirno dosega trofogeno cono. Nasprotno pa se nitrat izloča iz sistema v procesu denitrifikacije.

3.5 Matematične formulacije ekoloških procesov

Zaradi izjemne kompleksnosti naravnih sistemov je nemogoče, oz. nepraktično detajlno matematično opisati vse procese, ki se odvijajo v naravi. V tem poglavju podajamo formulacije bolj posplošenih procesov, ki se večinoma uporabljajo v ekoloških modelih jezer in s katerimi se večinoma dovolj natančno približamo realni situaciji. Te procese smo zajeli v knjižnici znanja za avtomatizirano modeliranje procesov v jezeru (glej tudi prilogo A).

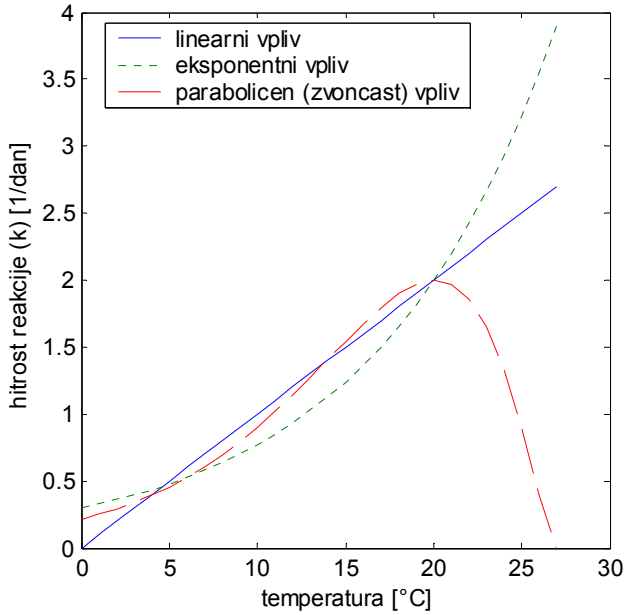
Pri izdelavi trofičnega modela jezera ločimo fizikalne in biogeokemijske procese. Med fizikalne prištevamo procese, ki prispevajo k obremenitvi sistema s hranili, t.j. dotok hranil s pritoki, iz prispevnega območja, s padavinami ter s komunalnimi izpusti in procese mešanja in transporta snovi znotraj sistema. Biogeokemijski procesi so rast primarnih producentov, respiracija, odmiranje, pašnja, dekompozicija, hidroliza, izločanje, itd.

3.5.1 Zunanji vplivi na procese v jezeru

Zunanji vplivi so gonilne funkcije (neodvisne spremenljivke), ki vplivajo na biokemijske procese. V tem poglavju bomo opisali vpliv temperature, svetlobe in zunanje obremenitve sistema s hranili.

Temperatura

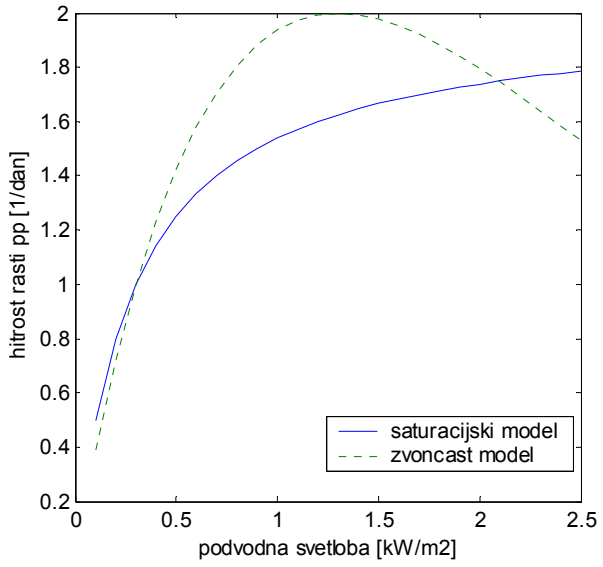
Temperatura vpliva na večino biokemijskih procesov. Večina modelov upošteva tri glavne kategorije vpliva temperature, to so (1) linearni vpliv, (2) eksponentni in (3) paraboličen (zvončasti) vpliv na procese, t.j. naraščanje hitrosti procesa do neke optimalne temperature, nato pa upadanje z naraščanjem temperature. Kot primer vzemimo vpliv temperature na hitrost rasti primarnih producentov. Slika 7 prikazuje vpliv temperature na maksimalno hitrost rasti fitoplanktona. Formulacije temperaturnih vplivov so zbrane v prilogi A.



Slika 7: Različni modeli vpliva temperature na hitrost rasti primarnih producentov (pp)

Svetloba

Svetloba je gonilna sila za rast primarnih producentov s tem pa posledično tudi za celotno prehranjevalno mrežo. Vpliv svetlobe je zajet preko treh faktorjev: dnevna variacija intenzitete svetlobe, upadanje svetlobe z globino v vodnem stolpcu in odvisnost hitrosti rasti od ambientalne (podvodne) svetlobe (npr. Chapra, 1997). Zadnje izražamo z različnimi modeli, ki so zbrani v prilogi A. Slika 8 prikazuje dva tipa modelov (1) saturacijski model in (2) model ki upošteva optimalno svetlobo (zvončast model). Nekateri avtorji (Talbot et al., 1991; Thebault in Salencon, 1993) upoštevajo kombiniran vpliv temperature in svetlobe na rast primarnih producentov (glej prilogo A).



Slika 8: Modeli za vpliv svetlobe. saturacijski model (polna črta) in zvončast model (črtkano)

Zunanja obremenitev sistema s hranili
Zunanja obremenitev prispeva h koncentraciji hranil v sistemu. Zajeta je preko izpiranja hranil iz prispevnega območja, padavin, dotokov s pritoki. Formulacija teh procesov, t.j. vtok, iztok, obremenitev iz prispevnih površin, ter obremenitev s padavinami je navedena v prilogi A.

3.5.2 Fizikalni procesi

Sem spadajo procesi mešanja in transporta snovi. Povezujejo stanja v sistemu med različnimi oddelki, kot je npr. kontakt med sedimentom in vodno maso. Med njih prištevamo: sedimentacijo, vtok in iztok hranil, difuzijo ali mešanje snovi. Formulacije teh procesov so podane v prilogi A.

3.5.3 Kemijski procesi

Med kemijske procese prištevamo transformacije anorganskih hranil iz ene v drugo obliko, kot sta recimo nitrifikacija (transformacija amonijaka, oz. amonijevega iona v nitratni ion) in denitrifikacija (transformacija nitrata (nitratnega iona) v elementarni dušik), hidroliza raztopljene organske snovi in dekompozicija neraztopljene (suspendirane) mrtve organske mase. Matematično jih večinoma formuliramo z kinetičnimi enačbami prvega reda, (18) in (19):

$$\frac{dC}{dt} = k(T) \cdot C \tag{18}$$

$$k(T) = k(T_{ref}) \cdot f(T) \tag{19}$$

kjer je *C* koncentracija snovi [masa/volumen], *k(T)* je hitrost reakcije, odvisna od temperature [1/čas], *k(Tref)* vrednost koeficienta hitrosti reakcije pri referenčni temperaturi in *f(T)* funkcija vpliva temperature na hitrost reakcije.

3.5.4 Bio-kemijski procesi

Rast primarnih producentov

Dinamiko rasti primarnih producentov večinoma formuliramo s tremi osnovnimi modeli. Eksponentni model predpostavlja konstantno rast (20). Vsi viri potrebni za rast so v neomejenih količinah in tudi zunanji vplivi so optimalni. Koeficient rasti je konstanta.

$$rast\_pp = \mu \cdot PP \tag{20}$$

kjer je PP koncentracija primarnega producenta [masa/volumen] in μ je koeficient rasti [1/čas]. Za razliko od eksponentnega modela, logistični model (Verhulst, 1845) predpostavlja omejeno naraščanje populacije. Omejitev predstavlja gostota populacije. V tem primeru koeficient rasti ni več konstanta, temveč ga formuliramo po enačbi (21):

$$\mu = \mu_{\max} \cdot PP \cdot \left(1 - \frac{PP}{PP_C}\right) \tag{21}$$

kjer je $\mu_{\max}$ koeficient maksimalne rasti in PP_C zgornja meja rasti.

Večina ekoloških modelov upošteva vpliv različnih dejavnikov na rast, kot so svetloba, temperatura in hranila. Ta vpliv lahko upoštevamo na različne načine v modelu rasti. Zelo pogosto predstavljmo skupni vpliv naštetih dejavnikov kot produkt vplivnih funkcij posameznih dejavnikov (22).

$$\mu = \mu_{\max}(T_{ref}) f_1(T) \cdot f_2(L) \cdot f_3(N,P,C) \tag{22}$$

kjer je $\mu_{\max}(T_{ref})$ hitrost rasti *pp* pri optimalnih pogojih in referenčni temperaturi, $f_1(T)$ funkcija vpliva temperature na rast, $f_2(L)$ funkcija vpliva svetlobe in $f_3(N,P,C)$ omejitvena funkcija koncentracij hranil.

Hranila kot omejitveni dejavniki za rast fitoplanktona

Skupno funkcijo za omejitev rasti zaradi hranil lahko izrazimo kot kombinacijo omejitvenih funkcij posameznih hranil. To lahko storimo na več načinov, kot prikazujejo enačbe (23), (24) in (25):

$$f(P,N,C) = \min\left[f(P), f(N), f(C)\right] \tag{23}$$

$$f(P,N,C) = f(P)f(N)f(C) \tag{24}$$

$$f(P,N,C) = \frac{f(P) + f(N) + f(C)}{n} \tag{25}$$

kjer so P, N, C anorganska hranila, t.j. fosfor, dušik in ogljik.

Trenutno, knjižnica znanja podpira le kombinacijo dejavnikov po (24). Omejitvena funkcija ima lahko vrednost od nič do ena. Ena pomeni, da hranilo ne omejuje rast fitoplanktona, nič pa pomeni, da ima hranilo tak vpliv, da rast preneha takoj.

Večinoma uporabljamo dva pristopa k modeliranju vpliva hranil na rast fitoplanktona. Prvi upošteva konstantno stehiometrično sestavo celic alg, ter bazira na Monodovi enačbi. Hitrost rasti je določena z zunanjimi koncentracijami hranil, t.j. s koncentracijami v vodnem stolpcu. Drugi pristop upošteva spremenljivo stehiometrično sestavo celic. Rast alg je upoštevana v dveh korakih: (1) asimilacija hranil iz okolice in (2) rast ali delitev celice. Asimilacija je odvisna tako od zunanje kot od notranje koncentracije, medtem ko je rast celic odvisna le od notranje koncentracije hranil. S takim pristopom (ločena procesa asimilacije in rasti) lahko modeliramo spremembe v celični sestavi s časom. Taki modeli lahko simulirajo tudi situacije, ko beležimo rast tudi pri nični zunanji koncentraciji hranil (ko so se mikroorganizmi preskrbeli z dodatno zalogo v času obilice hranil – luxury uptake, surplus internal quota).

Večina modelov se opira na prvi koncept modeliranja omejitvenega vpliva hranil, ki je tudi zajet v naši knjižnici znanja. Monodov model (Monod, 1949) za omejitveno funkcijo hranila N se glasi:

$$f_3(N) = \frac{N}{K_N + N} \tag{26}$$

kjer je K_N koncentracija hranila N pri $\mu_{\max}/2$ (polsaturacijska konstanta). Če želimo upoštevati več hranil kot omejitvene dejavnike, napišemo podobne izraze za vsako izmed hranil in jih kombiniramo po eni izmed formul (23), (24) ali (25). Nekaj podobnih, često uporabljanih, omejitvenih funkcij podajamo v prilogi A.

Rast sekundarnih producentov

Ta proces predstavlja interakcijo med plenom in plenilcem. Za plen predstavlja izgubo, medtem ko za plenilca pa rast. Bistvena razlika v formulaciji procesa rasti primarnega in sekundarnega producenta je ta, da primarni producent potrebuje vsa hranila (anorganska) hkrati. Če enega ni, potem ni rasti. Zato tudi modeliramo omejitev rasti zaradi hranil s produktom omejitvenih funkcij posameznega hranila ((24). Sekundarni producent pa lahko konzumira več vrst hrane (plena) ali pa samo eno. Če nekatere vrste ni, bo jedel drugo. V tem primeru je omejitveni faktor vsota hrane, ki jo organizem konzumira (27). Če upoštevamo selektivno prehranjevanje, potem F_T dobi obliko, kot kaže enačba (28).

$$F_T = \sum_{k=1}^{n} F_k \tag{27}$$

$$F_T = \sum_{k=1}^{n} pf_k \cdot F_k \tag{28}$$

kjer je F_T skupna koncentracija hrane, F_k je koncentracija posamezne vrste hrane k in p_{fk} je preferenčni faktor za vrsto hrane F_k.

V splošnem lahko formuliramo proces rasti sekundarnih producentov na dva načina. Prvi uporablja koeficient hitrost porabe plena C_g, kot kaže enačba (29) drugi pa koeficient filtracije vode C_f enačba (30), saj se npr. večina zooplanktona hrani na tak način. V obeh formulacijah upoštevamo korigirane maksimalne vrednosti koeficientov (C_{gmax} in C_{fmax}) s temperaturno funkcijo $f_1(T)$ in omejitveno funkcijo hrane $f_2(F_T)$. Formulaciji sta prikazani spodaj (29) in (30):

$$rast\_sp = C_{g\max} \cdot f_1(T) \cdot f_2(F_T) \cdot pred \tag{29}$$

$$rast\_sp = C_{f\max} \cdot f_1(T) \cdot f_2(F_T) \cdot pred \cdot plen \tag{30}$$

kjer je *plen* koncentracija plena [masa/volumen], *pred* koncentracija predatorja [masa/volumen] $C_{g\max}$ koeficient maksimalne hitrosti porabe plena [masa plena/(masa pred * čas)], $C_{f\max}$ maksimalna hitrost filtracije vode [m^3/(g pred*čas)].

Respiracija in izločanje

Procesa respiracije in izločanja štejemo pod izgube. Izločki primarnih in sekundarnih producentov bistveno vplivajo na kroženje hranil. V splošnem jih matematično formuliramo s kinetiko prvega reda, kjer je hitrost reakcije funkcija temperature in/ali fiziološkega stanja organizma. Veliko modelov vključuje le temperaturno odvisnost. Scavia (1980) in Recknagel (1980) uporabljata model, ki

povezuje koeficient respiracije s fiziološkim stanjem alg. Formulacije procesov za primarne in sekundarne producente so podane v prilogi A.

Umrljivost
Z naravno umrljivostjo primarnih producentov zajamemo poleg naravne smrtnosti še procese, kot so senčenje, razgradnja celic s strani bakterij (parazitizem), umrljivost pogojena z močnim upadom hranil, umrljivost zaradi ekstremnih okoljskih pogojev ali toksičnih substanc. Ponavadi ta proces vključujemo v model, ko ne modeliramo ostalih procesov izgub, kot je recimo usedanje. Formulacija procesa je lahko enostavna, kot je kinetična enačba prvega reda, ali pa kompleksnejša, kot je recimo uporaba Monodove saturacijske funkcije koncentracije alg za omejitev hitrosti umrljivosti (Nyholm, 1978). Formulacije koeficienta hitrosti umrljivosti primarnih producentov so prikazane prilogi A.

Umrljivost sekundarnega producenta (sp) običajno formuliramo z enostavnim modelom kinetike prvega reda, če v modelu nastopa tudi predator sekundarnega producenta. Če pa ta proces predstavlja zaključni člen v modelu, t.j. modelirana prehranjevalna veriga se zaključi s tem sekundarnim producentom, potem uporabljamo kompleksnejše izraze za formulacijo tega procesa. Bierman s sodelavci (Bierman et al., 1980) npr. uporabljajo kinetiko drugega reda, ko koncentracija sekundarnega producenta preseže določen nivo. Druge formulacije vključujejo kvadratno (npr. Steele in Henderson, 1981; Fasham, 1995), hiperbolično (npr. Frost, 1987; Fasham, 1993; Ross et al., 1994) in sigmoidno (Malchow, 1994) obliko formulacije procesa. Enačbe koeficienta hitrosti umrljivosti sekundarnih producentov so prikazane v prilogi A.

3.5.5 Kisikov model

Enostaven kisikov model v jezeru lahko zapišemo na naslednji način:

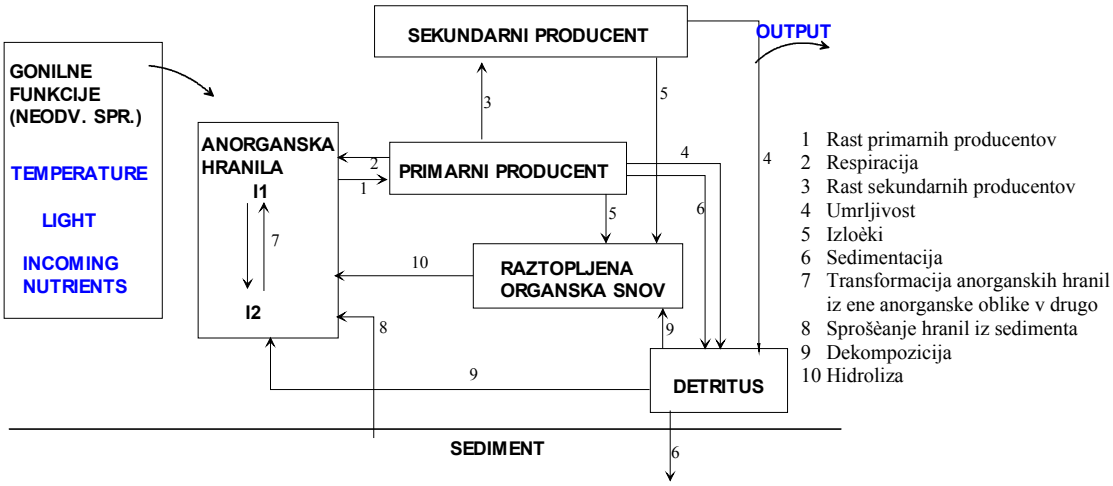
$$\frac{dO}{dt} = reaeracija - poraba + produkcija \tag{31}$$

Reaeracija predstavlja izmenjavo kisika med površino in vodno gladino. Poraba kisika v jezerih poteka zaradi sledečih procesov: mikrobiološka razgradnja raztopljene in suspendirane organske snovi (dekompozicija), oksidacija amonija (nitrifikacija), poraba v sedimentu, t.j. oksidacija usedle organske snovi in respiracija bentičnih organizmov ter dihanje vseh ostalih organizmov. Produkcija kisika se odvija v procesu fotosinteze. Matematična formulacija omenjenih procesov je podana v prilogi A.

4 Razvoj domenske knjižnice za modeliranje prehranjevalne verige v jezeru

Poudarek tega poglavja je razvoj knjižnice znanja o modeliranju prehranjevalnih mrež v jezerih v obliki generičnih procesov. Z uporabo formalizma, ki ga je razvil Todorovski (2003) smo zapisali generalizirano znanje o modeliranju jezer v domensko knjižnico. V prilogi A podajamo podrobnejši opis knjižnice, izpis celotne knjižnice pa je podan v prilogi A.1.

Znanje o modeliranju vodnih ekosistemov, ki smo ga opisali v prejšnjem poglavju, lahko generaliziramo, kot je kaže Slika 9. Prikazana je posplošena shema interakcij med spremenljivkami stanja v vodnih ekosistemih, ki jo uporablja večina modelov. Tipi sistemskih spremenljivk so prikazani v pravokotnikih, puščice pa ponazarjajo zveze med njimi. Te zveze predstavljajo procesne razrede. Tipi spremenljivk in procesni razredi so opisani v nadaljevanju.

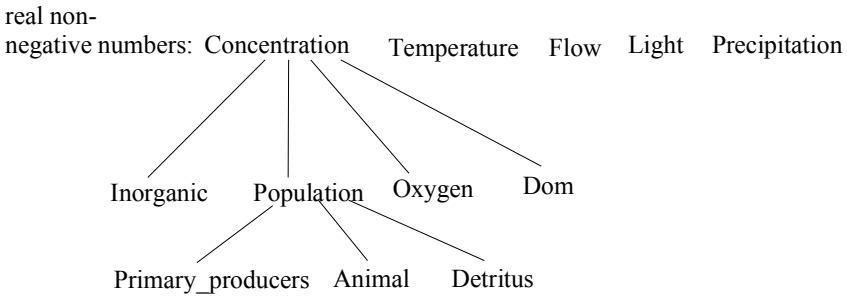


Slika 9: Generalizirana shema neodvisnih in sistemskih spremenljivk ter njihovih interakcij v vodnem ekosistemu

4.1 Formalizacija domenskega znanja

4.1.1 Taksonomija tipov spremenljivk

Taksonomija tipov spremenljivk obsega odvisne (sistemske) in neodvisne spremenljivke in je v skladu s spremenljivkami navedenimi v prejšnjem poglavju (slika) Shematično je taksonomija prikazana na Slika 10. Osnovni tipi so koncentracija (*Concentration*), Temperatura (*Temperature*), Pretok (*Flow*), svetloba (*Light*), in padavine (*Precipitation*). Tip *Concentration* ima štiri podtipe, *Inorganic*, ki predstavlja raztopljena anorganska hranila, *Population* (organska neraztopljena snov), *Dom* (raztopljena organska snov) in *Oxygen* (raztopljen kisik). Nadalje ima tip *Population* tri podtipe, t.j. *Primary_producers* (primarni producenti), *Animal* (sekundarni producenti) in *Detritus* (mrtva organska snov).



Slika 10: Shematični prikaz taksonomije tipov spremenljivk deklariranih v knjižnici

4.1.2 Taksonomija procesov

Večina procesnih razredov v knjižnici je shematično prikazana na sliki. Njihova organizacija v knjižnici pa je vendar nekoliko bolj kompleksna. Vzemimo za primer procesni razred Respiracija. Slika 9 ga prikazuje kot enak proces tako za respiracijo živali kot respiracijo primarnih producentov. Vendar pa je potrebno proces definirati kot dva ločena procesna razreda, t.j. respiracija primarnega producenta in respiracija sekundarnih producentov, saj se procesa nekoliko razlikujeta v svojih formulacijah. Tudi s strani uporabnika je to ugodno v primeru, da želi respiracijo primarnega producenta drugače formulirati (oz. omejiti iskanje v tem razredu) kot respiracijo živali. Podobno velja tudi za proces umrljivost.

Procesa rasti primarnih producentov in sekundarnih producentov je prav tako potrebno ločiti v dva procesna razreda. Formulacije teh dveh procesov se bistveno razlikujeta že zaradi narave prehranjevanja primarnih in sekundarnih producentov, kot smo opisali v poglavju 3. Todorovski (2003) je v svoji knjižnici uporabil en procesni razred, kar ima določene pomanjkljivosti – glej podrobneje v prilogi B, kjer je opisana razlika med knjižnicama.

Tabela 13 prikazuje opis definicije večine procesnih razredov iz domenske knjižnice. Podrazredi procesov niso prikazani. V prvem stolpcu je podan opis procesnega razreda. Drugi stolpec vsebuje imena procesnih razredov, medtem ko so v tretjem in četrtem stolpcu podatki o argumentih, oz. tipih spremenljivk, ki nastopajo v procesnem razredu. Iz tretjega stolpca lahko razberemo, koliko argumentov vsebuje določen proces in katerega tipa so, iz četrtega pa, ali je določen argument definiran kot množica. Podrobnejši opis procesov, kot tudi celotne knjižnice podajamo v prilogi A. Izpis celotne knjižnice pa je podan v prilogi A.1

Tabela 13: Opis definicij procesnih razredov v knjižnici

	Opis procesnega razreda	Ime procesnega razreda	Argumenti: tipi spremenljivk, ki nastopajo v procesnem razredu	Argument deklariran kot množica: da/ne
1	Izpiranje določene substance z pretokom	Outflow	Concentration Flow	ne ne
2	Obremenitev sistema zaradi dotoka substance s pretokom	Inflow	Concentration Concentration Flow	ne ne ne
3	Obremenitev s hranili iz prispevnega območja	Lin_load	Inorganic Inorganic Area	ne ne ne
4	Usedanje substance	Sedimentation	Concentration Temperature	ne da
5	Mešanje dveh substanc (difuzija)	Diffusion	Concentration Concentration	ne ne
6	Rast primarnega producenta	PP_growth	Primary_producer Inorganic Temperature Light	ne da da da
7	Rast sekundarnega producenta (interakcija plen-plenilec)	Feeds_on	Animal Population Temperature	ne da da
8	Respiracija primarnega producenta	Respiration_PP	Primary_producer Inorganics Temperature Light	ne da da da
9	Respiracija sekundarnega producenta	Respiration_A	Animal Temperature	ne da
10	Umrljivost primarnega producenta	Mortality_PP	Primary_producer Inorganic Temperature Light	ne da da da
11	Umrljivost sekundarnega producenta	Mortality_A	Animal Temperature	ne da
12	Izločki sekundarnega producenta	Excretion_A	Animal Temperature	ne da
13	Dekompozicija suspendirane mrtve organske snovi	Decomposition	Detritus	ne

4.1.3 Kombinatorne sheme

V kombinatornih shemah (combining scheme) zapišemo ustreno kombinacijo procesov za vsako odvisno spremenljivko. Ta kombinacija ustreza masni bilanci spremenljivke. V knjižnici imamo deklariranih šest tipov odvisnih spremenljivk, torej imamo šest kombinatornih shem oz. masnih bilanc. Kombinatorno shemo primarnega producenta kaže Tabela 14.

Tabela 14: Kombinatorna shema primarnega producenta

```
combining scheme Lake(Primary_producer pp)
        time_deriv(pp) =
                + sum({food, ts, ls}, true, PP_growth(pp, food, ts, ls))
                - sum({ns,ts,ls}, true, Respiration_PP(pp,ns,ts,ls))
                - sum({ns,ts,ls}, true, Mortality_PP(pp, ns,ts,ls))
                - sum({}, true, Outflow(pp))
                - sum({ts}, true, Sedimentation(pp,ts))
                + sum({pp1}, true, Diffusion(pp,pp1))
                - sum({pp1}, true, Diffusion(pp1,pp))
                - sum({a, food, ts}, pp in food, Feeds_on(a, food, ts))*Food_pref(pp)
```

4.2 Prikaz splošnosti zajetega znanja v knjižnici

Posplošeno znanje na osnovi procesov, oz. posameznih gradnikov za modeliranje leteh, naj bi omogočalo poenoten modularni pristop k gradnji modelov različnih vodnih ekosistemov. Z uporabo predznanja v knjižnici smo zapisali več znanih in uveljavljenih modelov vodnih ekosistemov, kot so enostavni Vollenweider-jev model (Vollenweider, 1968), Imboden-ov (Imboden, 1974) in SALMO (Bendorf, 1979; Recknagel, 1980). – priloga A. Specifikacijo sistema s katero zgeneriramo gramatiko modelov iz knjižnice podajamo tako, kot smo opisali v poglavju 3. Razvidno je, da za pravilno zapisano specifikacijo sistema potrebujemo znanje o tipih spremenljivk in procesnih razredih definiranih v knjižnici. Pomagamo si lahko s Tabela 13. Vrstica 6 npr., podaja definicijo procesnega razreda rast primarnih producentov. Ime procesa je *PP_growth* in vsebuje štiri argumente. Prvi je spremenljivka tipa Primary_producer, ostali pa so množice tipov *Inorganic, Temperature in Light*. Specifikacija procesa:

process PP_growth(phyto1, {ps}, {temp}, {llight}) **growth,**

opisuje rast primarnega producenta *phyto1*. Rast omejuje hranilo *ps*, na proces pa vplivata tudi temperatura *temp* in svetloba *light*. Ker so *ps temp in light* deklarirani kot množice, lahko proces specificiramo tako, da je v množici več spremenljivk istega tipa ali pa nobena. To bo vplivalo na končno formulacijo procesa. Recimo, da želimo specificirati rast primarnega producenta *(phyto1)* tako, da ga omejujeta dve hranili *(ps in ns)*, na proces pa vpliva le temperatura *temp*. Svetlobo iz določenega razloga ne želimo vključiti v formulacijo procesa (npr. svetlobe nimamo merjene). V tem primeru zapišemo:

process PP_growth(phyto1, {ps, ns}, {temp}, {}) **growth.**

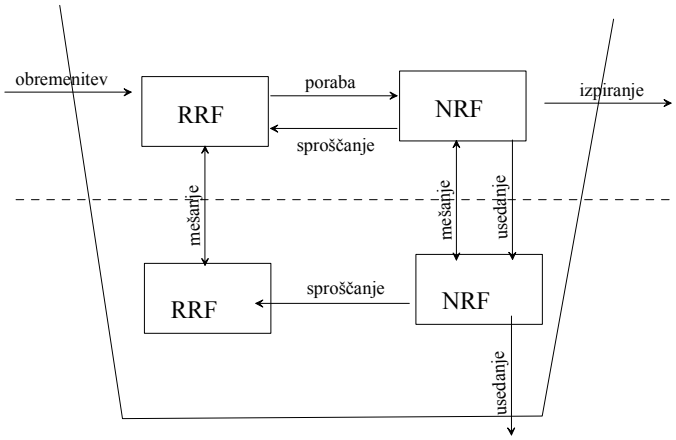
V nadaljevanju pdajamo kratek opis omenjenih modelov, ki smo jih s specifikacijo sistemov, za katere so izdelani, zapisali z uporabo generičnih procesov v knjižnici.

Vollenweiderjev model je eden izmed prvih modelov za modeliranje evtrofikacije v jezerih. Upošteva eno odvisno spremenljivko, t.j. totalni fosfor kot pokazatelj evtrofnosti. Enačba modela se glasi (32):

$$V\frac{d[P]}{dt} = P_{tot} - V \cdot K_{sed} \cdot [P] - Q \cdot [P] \tag{32}$$

kjer pomeni *[P]* koncentracija totalnega fosforja v sistemu, P_{tot} totalna zunanja obremenitev in K_{sed} hitrost usedanja v sediment. Model opisuje dinamiko totalnega fosforja z upoštevanjem treh procesov, t.j dotok fosforja zaradi zunanje obremenitve, usedanje in izpiranje oz sistema.

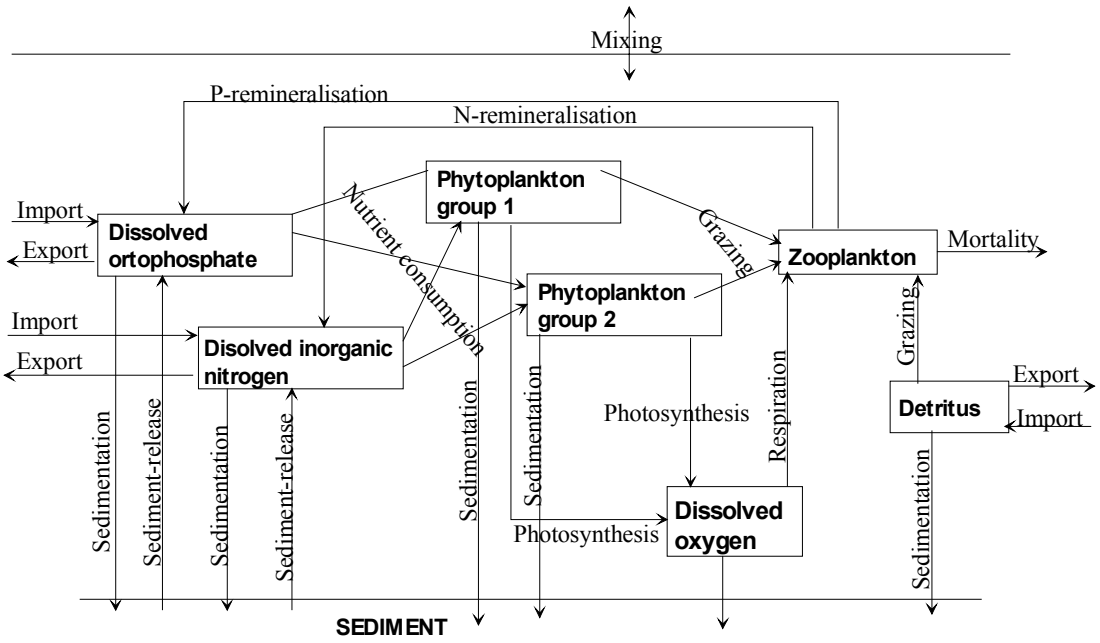
Izboljšave (razširitve) modela so šle v smeri modeliranja več odvisnih spremenljivk (npr. anorganski in organski fosfor), kakor tudi upoštevanja razslojevanja jezera, ter izmenjave hranil s sedimentom. O'Melia (1972), Imboden (1974) in Snodgrass (1974) so prvi podali smernice za simulacijo fosforja v takem sistemu. Konceptualni model prikazuje Slika 11.



Slika 11: Konceptualni model fosforjevega kroga (Imboden, 1974)

Karakteristike modela so sledeče (1) Fosfor je razdeljen na dve komponenti: raztopljen reaktiven fosfor (RRF) in neraztopljen reaktiven fosfor (NRF), (2) jezero je razdeljeno v dva dobro premešana dela konstantne debeline, (3) za opis procesov transporta in kinetike so uporabljene diferencialne enačbe prvega reda in (4) model ima štiri odvisne spremenljivke, torej imeli bomo štiri diferencialne enačbe – za vsako spremenljivko modela po eno bilančno enačbo - s toliko členi, kolikor je procesov, ki vplivajo na posamezno spremenljivko. Model je doživel nekaj izboljšav. Imboden in Gachter (1978) ter Imboden (1979) sta zamenjala izraze kinetike prvega reda z Monodovo kinetiko. Naša knjižnica podpira vse variante (glej prilogo A).

Bendorf (1979) in Recknagel (1980) sta izdelala relativno kompleksen model SALMO (Slika 12). Model upošteva razslojevanje jezera na epi- in hipolimnij. Vsak sloj vsebuje sedem spremenljivk stanja, t.j dve anorganski hranili (na sliki označeni kot *Dissolved ortophosphate in Dissolved inorganic nutrient*), dve vrsti fitoplanktona (*Phytoplankton group1 in Phytoplankton group 2*), ena vrsta zooplanktona (*Zooplankton*), detritus (*Detritus*) in kisik (*Dissolved oxygen*).



Slika 12: Koncept modela SALMO (Bendorf, 1979; Recknagel, 1980). V kvadratih so zapisane spremenljivke stanja: raztopljeni fosfor (Dissolved orthophosphate), nitrat (Dissolved inorganic nitrogen), dve skupini fitoplanktoma (Phytoplankton group 1 in phytoplankton group 2), zooplankton (Zooplankton), mrtva suspendirana snov (Detritus) in raztopljen kisik (Dissolved oxygen).

Opisane modele je torej mogoče rekonstruirati z knjižnico znanja, razvito v okviru te naloge. Specifikacija sistemov za vse tri modele je natančno opisana v prilogi A. To nakazuje na splošnost zajetega znanja o modeliranju prehranjevalnih verig v jezeru.

5 Aplikacija Domenske knjižnice in LAGRAMGEa na realnih podatkih

Cilj aplikacij je izdelati čimboljše modele jezer z uporabo kombiniranega pristopa k modeliranju, t.j z vpeljavo domenskega znanja v postopek indukcije enačb. Orodje LAGRAMGE smo aplicirali na štirih domenah: jezero Glumsø, Beneška Laguna, jezero Kasumigaura in Blejsko jezero. Uporabljali smo knjižnico znanja razvito v okviru te naloge. V nadaljevanju podajamo opis domen in podatkov, ki so nam bili na razpolago za vsako domeno.

5.1 Opis domen, podatkov in eksperimentov

5.1.1 Beneška Laguna

Beneška laguna je v povprečju dolga 50 km in 10 km široka. Celotna površina meri 550 km^2 ter v povprečju dosega globino manj kot 1 m. Letni vnos hraniv znaša 7 000 t dušika, fosforja pa 1 400 t (Bendoricchio et al., 1994; Bendoricchio et al., 1993; Coffaro in Sfriso, 1997; Coffaro in Bocci, 1997) kar znese letno na kvadratni meter lagune 13 g N in 2,5 g P. Te obremenitve daleč presegajo količino hraniv, ki jih laguna še prenese ter povzročajo njeno distrofično stanje, ki se odraža z prekomerno zarastjo alg, predvsem makroalgo *Ulva rigida*. Domnevno, je glavni razlog odmiranja (izgub) te alge lastno preprečevanje dostopa svetlobe (senčenje spodnjih slojev) zaradi prekomerne rasti in ne pašnja zooplanktona ali drugih živali. Posledica masovnega odmiranja alge je poraba kisika za dekompozicijo mrtve mase, kar nadalje vpliva na višje živali ter povzroča smrad in slab izgled lagune.

Podatkovna baza vsebuje meritve na štirih lokacijah, t.j. 0, 1, 2 in 3 (Coffaro et al., 1993). Vzorčevanje na lokaciji 0 je potekalo tedensko, leta 1985/86, na preostalih lokacijah pa prav tako tedensko, leta 1990/91. Merjeni so bili naslednji parametri: dušik v obliki amonija (*nh*) v [mg/l], nitratni dušik (*no*) v [mg/l], ortofosfat (*ps*) v [mg/l], raztopljen kisik (*DO*), [% saturacije], temperatura (*temp*) v [°C] in biomasa alg (*biomass*) v [suha teža g/m^2].

Iz danih podatkov in z uporabo ekspertnega znanja smo poskušali odkriti model za napoved biomase alg (priloga B). Podano ekspertno znanje v postopek odkrivanja enačb vsebuje deklaracijo merjenih spremenljivk v sistemu ter deklaracijo naslednjih procesov: rast primarnih producentov (PP_growth), respiracija (Respiration_PP), usedanje (Sedimentation) in umrljivost (Mortality_PP). Dobljene enačbe smo primerjali z enačbami dobljenimi z uporabo enostavnejše knjižnice znanja (Todorovski, 2003) (glej prilogo B).

5.1.2 Jezero Glumsø

Površina jezera Glumsø (Jørgensen et al., 1986) meri 266,000 m^2. Je plitvo jezero s povprečno globino blizu 2 m. Nekaj let se je v jezero stekala odpadna voda naselja s 3 000 prebivalci, očiščena do druge stopnje, t.j. biološko je bil odstranjen organski

ogljik ne pa tudi dušik in fosfor. Dodatna obremenitev z dušikom in fosforjem je prispevna površina jezera, ki meri 10.9 km^2 in je pretežno agrarnega značaja. Visoke obremenitve s hranili (dušik in fosfor) povzročajo hipereutrofno stanje jezera. Jezero ne vsebuje podvodne vegetacije, najverjetneje zaradi slabe prosojnosti in deficita kisika.

Za jezero Glumsø smo imeli na razpolago dva niza podatkov (A in B). Podatkovni niz A vsebuje štirinajst meritev v dveh mesecih in sicer dnevni pretok skozi jezero, temperatura, raztopljena hranila (nitrat in fosfor), totalna biomasa fitoplanktona v [mg suha teža /l] in biomasa zooplanktona [mg suha teža /l]. Zaradi mnogo premajhne količine meritev za potrebe avtomatskega modeliranja, so bila izvedena dodatna procesiranja za pridobitev ustreznega podatkovnega niza (Kompare, 1995). Grafi časovnih odvisnosti obstoječih meritev so bili posredovani trem ekspertom, da bi ocenili potek določene spremenljivke v času med dvema meritvama. Tako so bile dobljene zvezne krivulje meritev, ki se lahko smatrajo kot dodatni vir zanesljivih podatkov.

Na tako dobljenem podatkovnem nizu smo izvedli eksperiment odkrivanja enačbe fitoplanktona. Rezultat smo primerjali z enačbo dobljeno z uporabo enostavnejše knjižnice (Todorovski, 2003). Postopek in rezultati so opisani v prilogi B.

Podatkovni niz B vsebuje dnevne meritve v obdobju od aprila 1973 do aprila 1974 (meritve 73/74) ter od oktobra 1974 do oktobra 1975 (meritve 74/75). Merjeni so sledeči parametri: pretok skozi jezero, raztopljen anorganski dušik (*ns*), ortofosfat (*ps*), fitoplankton (*phyto*) merjen kot Chl-a [mg/l], zooplankton (*zoo*) v [mg suha teža /l], temperatura (*temp*) in svetloba (*light*) v [J/cm^2*dan].

Ekspertno znanje potrebno za odkrivanje modela z LAGRAMGE-om ter njegov vnos v program (specifikacija opazovanega sistema) je razviden iz priloge D. Odkrivali smo model za fitoplankton na podatkih 74/75, medtem ko je bila validacija modela opravljena na meritvah 73/74.

5.1.3 Jezero Kasumigaura

Jezero Kasumigaura (Bobbin in Recknagel, 2001; (Wei et al., 2001) je plitvo jezero s povprečno globino 4 m. Volumen jezera znaša 800 mio m^3, površina pa 220 km^2. Kot večina plitvih jezer se tudi tu pojavlja problem evtrofikacije in cvetenja alg, ki ga najpogosteje povzroča vrsta *Microcystis*. (http://www.ilec.or.jp/database/asi/asi-35.html).

Podatki o jezeru (Recknagel, 2004) vsebujejo meritve od 1986 to 1992. Merjeni podatki so temperatura vode (*temp*), globalna radiacija (*light*), raztopljen fosfor (*ps*), nitrat (*no3*), amonij (*nh*) silicij (*silica*), totalni fitoplankton, merjen kot chl-a (*chla*), vrste fitoplanktona, t.j. *Microcistis, Oscillatoria, Scenedesmus in Synedra rumpens* merjeno v [Št. celic/l] in vrsta zooplanktona Cladocera *(clad*), merjeno v [Št. os/l]. Vse podatke razen temperature in svetlobe smo dobili kot linearno interpolirane podatke med dejanskimi meritvami. Domnevamo, da je bila pogostost meritev enkrat mesečno. Zooplankton je merjen le do leta 1989. Podrobnejša analiza

podatkov, ki je bila vodilo za postavitev eksperimentov, je razvidna iz priloge C. Z uporabo podatkov in vnešenim ekspertnim znanjem v postopek odkrivanja enačb (glej prilogo C) smo izvedli naslednje eksperimente za odkrivanje modela za skupni fitoplankton, oz. Chl-a:

(1) Identifikacija dinamike fitoplanktona za vsako leto z učenjem modelov na podatkih, pripravljenih za vsako leto. Pri tem eksperimentu smo želeli preveriti, ali se dinamika iz leta v leto ponavlja. Še zlasti nas je zanimalo, katera hranila omejujejo rast fitoplanktona in ali se le-ta ponavljajo iz leta v leto. Ali potrebujemo za opis te domene različne strukture modelov ali pa samo različne parametre in enako strukturo. Vsakega izmed odkritih modelov, naučenega na podatkih enega leta, smo validirali na preostala leta. S tem smo preverili, ali morda obstaja reprezentativno leto. Vpliv zooplanktona ni bil upoštevan v tem eksperimentu.

(2) Odkrivanje modela z učenjem iz celotnega podatkovnega niza, oz. na podatkih od 1986 do 1991, leto 1992 smo uporabili za validacijo. S tem eksperimentom smo odgovorili na sledeča vprašanja: Ali dolžina učnega niza vpliva na tovrstno modeliranje in kako? Ali je bolje najti eno reprezentativno leto za učenje ali pa se učiti na celotnem podatkovnem nizu, čeprav lahko vsebuje veliko šuma?

(3) V tretjem eksperimentu smo vključili vpliv zooplanktona oz. pašnje zooplanktona na fitoplankton, pri odkrivanju modela za fitoplankton (Chl-a). Za učenje smo uporabili podatkovni niz od 1986 do 1988, medtem ko smo model validirali na podatkih iz leta 1989. Razlogi za krajši podatkovni niz so navedeni zgoraj in v prilogi C.

5.1.4 Blejsko jezero

Blejsko jezero je tipično subalpsko jezero glacialno-tektonskega izvora. Zaseda površino 1.4 km^2 z maksimalno globino 30.1 m povprečno pa 17.9 m (Sketelj in Rejic, 1958; Rismal, 1980; Remec-Rekar, 1995). Potopljen greben v smeri sever-jug pri blejskem otoku deli jezero v dve kotanji – zahodna in vzhodna. Volumen vzhodne kotanje znaša $17.5*10^6$ m^3, maksimalna globina pa 24 m. Volumen zahodne kotanje meri $8.2*10^6$ m^3, maksimalna globina 30 m. Monitoring blejskega jezera se izvaja v okviru Slovenskega nacionalnega programa od 1975. Podatki, pridobljeni pri MOPE, ARSO, vsebujejo meritve (od 1987 do 2002) fizikalnih, kemijskih in bioloških parametrov. Vendar pa lahko štejemo kot konsistentne in ustrezne za indukcijo modelov le tiste od leta 1995 do 2002. Pogostost meritev znaša enkrat mesečno. Vzorčevano je na dveh lokacijah (vzhodna in zahodna kotanja) in sicer vsaka dva metra od površine do dna. Podatki, ki smo jih uporabili v eksperimentih so sledeči: temperatura *(temp)*, svetloba *(light)*, pretoki in raztopljeni fosfor v pritokih Krivica *(q_krivica in ps_krivica)*, Mišca *(q_misca in ps_misca)* in Radovna *(q_radovna in ps_radovna)*, iztoki iz jezera *(q_jezernica in q_natega)*, raztopljena anorganska hranila v jezeru, t.j. fosfor *(ps)*, nitratni dušik *(no)* in silicij *(silica)*, skupna biomasa fitoplanktona *(phyto)* in vrsta zooplanktona *(daph)*.

Priprava podatkov in vnos ekspertnega znanja v postopek odkrivanja modelov sta opisana v prilogi E. Ekspertno znanje obsega deklaracijo spremenljivk v sistemu ter procesov pomembnih za opis dinamike sistema.

Izvedli smo tri eksperimente. Najprej smo odkrivali model, ki ustrezno opisuje dinamiko fitoplanktona skozi več let. Identifikacija modela je bila izvršena na podatkih od 1995 do 2001 (podatki za učenje). Leto 2002 je bilo uporabljeno za validacijo. Dobljeni model fitoplanktona se je slabo prilegal meritvam. Zato smo predpostavili, da se struktura jezera preveč spreminja iz leta v leto, da bi ga lahko opisali z enim modelom. To hipotezo smo preverili v drugem eksperimentu, t.j. identifikacija sistema (model za fitoplankton) ločeno za vsako leto. Za vsako leto smo torej dobili drug model. Vsi modeli so se dobro prilegali meritvam, razen modela za leto 1996. Za to leto smo v tretjem eksperimentu odkrivali model za opis osnovne prehranjevalne verige, t.j hranilo-fitoplankton-zooplankton. Zaradi zahtevne nelinearne optimizacije, ki bi nastopila v tej nalogi (optimizacija treh enačb) smo identifikacijo oz. prostor možnih rešitev za ta model močno omejili (glej prilogo E).

5.2 Rezultati

5.2.1 Primerjava modelov dobljenih z enostavno in kompleksno knjižnico

Tu smo primerjali rezultate ne podatkovnih bazah Glumsø A in Beneška Laguna. Todorovski (2003) je uporabil enostavno knjižnico za izgradnjo modelov ki napovedujejo koncentracijo fitoplanktona. Z uporabo knjižnice zgrajene v okviru te naloge smo na istih podatkih zgradili nove modele, ki se razlikujejo po strukturi in natančnosti. Rezultati so podani v prilogi B. Za jezero Glumsø smo imeli še dodatni niz podatkov, kot smo opiusali v prejšnjem poglavju. Na tem nizu smo uspešno odkrili model fitoplanktona in ga tudi validirali. Rezultati so razvidni iz priloge D.

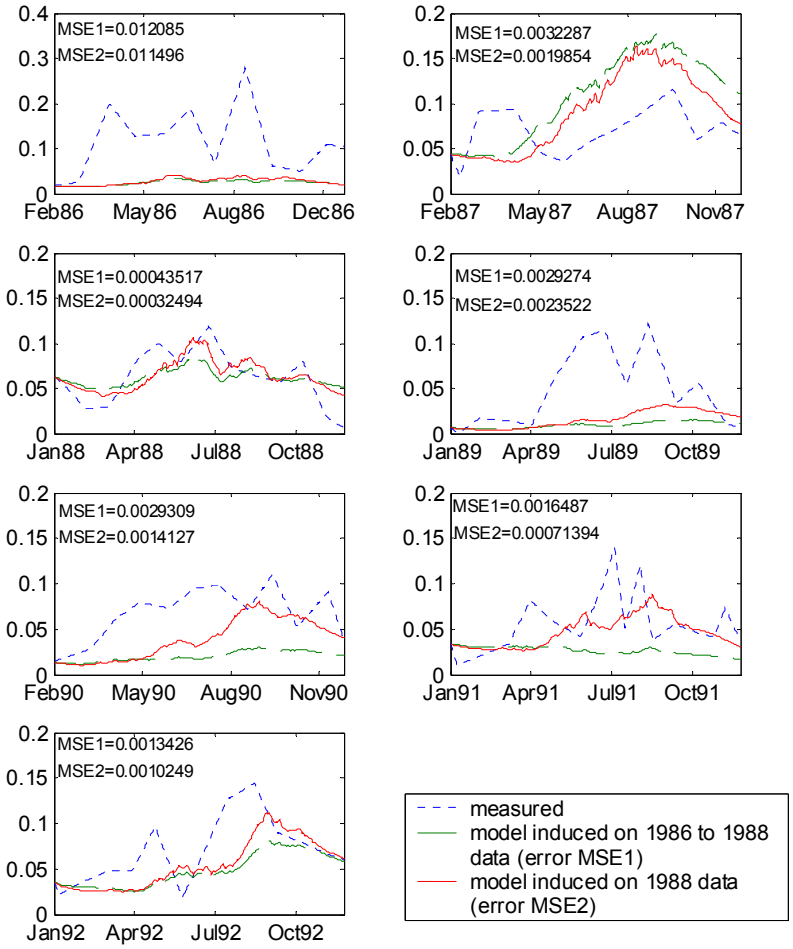
5.2.2 Jezero Kasumigaura

LAGRAMGE je v prvem eksperimentu uspešno odkril modele za vsako leto. Pri validaciji modelov na ostala leta (na nevidenih podatkih), se je kot najuspešnejši izkazal model odkrit na podatkih iz leta 1988 (33). Imena spremenljivk v enačbi so opisana v poglavju 5.1.3.

$$\frac{dchla}{dt} = chla \cdot 0.09 \cdot \frac{ps}{ps+0} \cdot \frac{no3}{no3+0} \cdot \frac{silica}{silica+0.022} \cdot \frac{temp}{10.8} \cdot \frac{light}{light+200} - chla \cdot 0.022 \cdot 1.11^{(temp-18.8)} - \\ - chla \cdot 0.01 \cdot \frac{temp}{7.2} - chla \cdot \frac{0.05}{5} \tag{33}$$

V drugem eksperimentu smo odkrivali podoben model, vendar z učenjem na celotnem podatkovnem nizu. Zdaj je LAGRAMGE odkril nekoliko drugačen model.

Za razliko od prejšnjega ta izključuje silicij kot omejitveno hranilo (priloga C). Primerjava simulacij z obema modeloma prikazuje Slika 13.



Slika 13: Primerjava simulacij z meritvami (pikčasta črta) dveh modelov, odkritih na podatkih jezera Kasumigaura (1) iz leta 1988 (polna črta) in (2) podatkih od 1986 do 1991 (črtkano)

V zadnjem eksperimentu smo preverjali kako zooplankton vpliva na dinamiko fitoplanktona. V tem poskusu smo izvedli tudi učenje na podatkih enega leta. LAGRAMGE ni nasel reprezentativnega modela (leta), ki bi uspešno simuliral fitoplankton v ostalih letih. Vendar pa je uspešno odkril modele (ki vključujejo vpliv zooplanktona) za vsako leto posebej. Modeli se nekoliko razlikujejo od tistih odkritih v prvem eksperimentu. Rezultati so podani v prilogi C.

5.2.3 Blejsko jezero

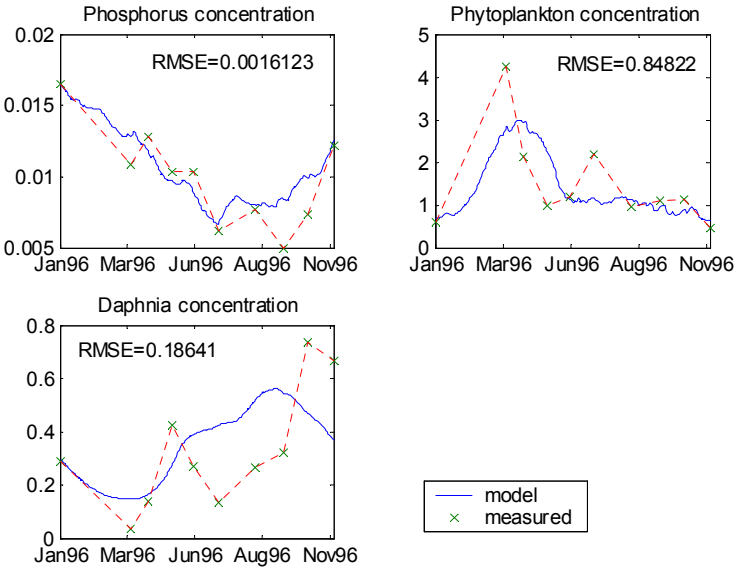
Rezultati na Blejskem jezeru vključujejo model za skupno biomaso fitoplanktona pri učenju na podatkih od 1995 do 2001, modele fitoplanktona naučene ločeno na podatkih vsakega leta od 1995 do 2002 in njihovo vačlidacijo na nevidenih podatkih, ter model treh enačb za osnovno prehranjevalno verigo (hranilo-fitoplankton-zooplankton). Rezultati so prikazani v prilogi E. V nadaljevanju prikazujemo model

treh enačb odkrit na podatkih iz leta 1996 (34), (35) in (36). Simulacijo modela kaže Slika 14.

$$\frac{dps}{dt} = ps\_krivica \cdot \frac{q\_krivica}{7 \cdot 10^6} + ps\_misca \cdot \frac{q\_misca}{7 \cdot 10^6} + ps\_radovna \cdot \frac{q\_radovna}{7 \cdot 10^6}$$
$$- ps \cdot \frac{q\_jezernica}{7 \cdot 10^6} - ps \cdot \frac{q\_natega}{7 \cdot 10^6} + 0.0022 \cdot phyto^2 \cdot 0.072 \frac{temp - 2.7}{19.7 - 2} +$$
$$0.07 \cdot daph \cdot 0.0026 \cdot \frac{temp}{12.3} - 0.0023 \cdot phyto \cdot 0.21 \cdot \frac{ps}{ps + 0.00042} \cdot \frac{temp}{16.7} \cdot \frac{light}{170} \cdot \exp(1 - \frac{light}{170}) \quad (34)$$

$$\frac{dphyto}{dt} = phyto \cdot 0.21 \cdot \frac{ps}{ps + 0.00042} \cdot \frac{temp}{16.7} \cdot \frac{light}{170} \cdot \exp(1 - \frac{light}{170}) - phyto^2 \cdot 0.072 \frac{temp - 2.7}{19.7 - 2} -$$
$$- phyto \cdot \frac{0.5}{10} \cdot \frac{temp - 2}{18 - 4} - daph \cdot 0.5 \cdot \frac{temp - 2.6}{18 - 4} \cdot (1 - \exp(-0.58 \cdot phyto)) \cdot 0.56 \cdot phyto \quad (35)$$

$$\frac{ddaph}{dt} = 0.14 \cdot daph \cdot 0.5 \cdot \frac{temp - 2.6}{18 - 4} \cdot (1 - \exp(-0.58 \cdot phyto)) \cdot 0.56 \cdot phyto -$$
$$- daph \cdot 0.026 \cdot \frac{temp}{12.3} - 0.01 \cdot \frac{daph^2}{0.001 + daph} \quad (36)$$



Slika 14: Simulacija modela treh enačb, odkritega na podatkih Blejskega jezera iz leta 1996. Levo zgoraj: koncentracija fosforja, desno zgoraj: koncentracija fitoplanktona in levo spodaj: koncentracija *Daphnie hialine*.

6 Diskusija

6.1 Vpliv ekspertnega znanja na iskanje ustreznih modelov

V tej nalogi smo uporabljali orodje za avtomatizirano modeliranje (LAGRAMGE), ki omogoča kombiniran pristop h gradnji modelov, t.j. indukcija iz podatkov z upoštevanjem ekspertnega znanja. Domensko znanje vnesemo preko specifikacije opazovanega sistema. Na ta način kontroliramo prostor možnih struktur modelov, ki so zapisane v t.i. gramatiki modelov. Bolj ko definiramo sistem manjši bo prostor možnih struktur (manjša gramatika) in obratno.

Gramatika modelov je največja, če specificiramo samo spremenljivke stanja v sistemu. V tem primeru bo LAGRAMGE iskal take modele za posamezno spremenljivko stanja, ki bodo vsebovali vse procesne razrede definirane v knjižnici znanja, ki lahko vplivajo na to spremenljivko. Iskanje lahko omejimo tako, da specificiramo procese (oz. procesne razrede), ki so (po mnenju eksperta) relevantni za to spremenljivko. Zdaj se bo iskanje omejilo na ustrezno formulacijo določenega procesa v kombinatorni shemi (masni bilanci) posamezne spremenljivke znotraj procesnega razreda (npr. iskanje ustrezne formulacije procesa Growth_PP, oz. rast primarnega producenta). Iskanje lahko omejimo tudi znotraj procesnega razreda s tem, da določimo željeno formulacijo posameznega procesa. V tem primeru je struktura modela popolnoma določena in LAGRAMGE izvede le kalibracijo parametrov na podani podatkovni niz. Teoretično lahko določimo tudi vrednost parametrov, kar bi pomenilo močno poseganje v metodo optimizacije (nični prostor iskanja, če določimo vse parametre).

Tako omejevanje prostora rešitev je prikladno zaradi same narave problema. Zavedati se je treba, da je prostor modelov, ki opišejo določeno domeno lahko zelo velik, kar pogojuje veliko računsko zahtevnost optimizacijske metode. Naj navedemo (Tabela 15) samo nekaj primerov zgeneriranih gramatik oz. število modelov za določeno specifikacijo, ki jih je potrebno optimizirati, in računske čase za izvršeno optimizacijo. Razvidno je, da že odkrivanje modela ene enačbe zahteva ogromno računskega časa. Seveda je to odvisno od specifikacije opazovanega sistem in kompleksnosti knjižnice znanja. Opažamo tudi, da je računski čas za odkrivanje enako kompleksnega modela na različnih podatkih iste domene različen. Zaradi tega, smo pri odkrivanju modela treh enačb na blejskih podatkih izkoristili možnost krčenja prostora rešitev (glej prilogo E). To smo storili tako, da smo odkrivali najprej eno enačbo, nato pa procese ki nastopajo v obeh enačbah (in smo jih že odkrili v prvi) upoštevali kot že poznane in naprej odkrivali samo še neznane procese. Tako smo našo nalogo simultanega odkrivanja modela treh enačb (oz treh spremenljivk stanja) prevedli na odkrivanje modela ene spremenljivke stanja (3x). Na ta način smo postopek bistveno pospešili, vendar pa se zavedamo, da to ni pravilna optimizacija z vsemi prostostnimi stopnjami.

Tabela 15: Število možnih modelov za posamezne primere in potreben računski čas za njihovo optimizacijo

Domena	Podatkovni niz	Št. modelov v gramatiki	Rač čas [ure]
Jezero Glumsø, nova knjižnica	podatkovni niz A	3 528	7.7
Jezero Glumsø, knjižnica Todorovski (2003)	podatkovni niz A	256	
Jezero Glumsø	podatkovni niz B	27 216	11.4
Beneška laguna	lokacija 0	3 240	8.3
Jezero Kasumigaura: eksperiment brez upoštevanja zooplanktona	Leta: 86-91 Leto: 86 Leto: 87 Leto: 88 Leto: 89 Leto: 90 Leto: 91 Leto: 92	18 144 18 144 18 144 18 144 18 144 18 144 18 144 18 144	83 39 43 37 34.5 36 26 37

6.2 Diskusija knjižnice znanja

Prava vrednost opisanega pristopa k izgradnji modelov, t.j. avtomatizirano modeliranje z upoštevanjem domenskega znanja, se izkaže, če imamo izčrpno in konsistentno knjižnico domenskega znanja. V okviru te naloge smo zgradili ontologijo za modeliranje vodnih ekosistemov. Gruber (1993) podaja zelo kratko definicijo ontologije, t.j. ontologija je eksplicitna in formalna specifikacija konceptualizacije (neke domene). Osnovni namen pa je uporaba in izmenjava znanja, ki ga ontologija pokriva.

Vodni ekosistem smo konceptualizirali preko spremenljivk, ki tu nastopajo in relacij, ki te spremenljivke povezujejo. To so tudi osnovni gradniki formalizma naše ontologije. Znanje je torej formalizirano v obliki taksonomij (hierarhično) tipov spremenljivk in procesnih razredov (relacije med spremenljivkami), kar omogoča generičnost zajetega znanja. Ključen element formalizma ontologije je znanje o ustrezni kombinaciji procesnih razredov v model celotnega vodnega ekosistema. Taka formalizacija znanja omogoča (1) enotni modularni pristop h gradnji modelov in (2) lažjo izmenjavo znanja med eksperti.

Splošnost zajetega znanja v knjižnici, oz. naše ontologije smo demonstrirali tako, da smo z ustrezno specifikacijo sistema zgenerirali znane modele. Na ta način smo pokazali, da naša knjižnica vsebuje modele od zelo enostavnih, kot je Vollenweiderjev model (ki vsebuje le eno enačbo) preko zmerno kompleksnih kot je Imbodenov model do razmeroma kompleksnih kot je SALMO (Bendorf, 1979; Recknagel, 1980). Imbodenov model in SALMO sta dvoslojna modela, t.j. upoštevata fizikalno razdelitev jezera na epi- in hipolimnij. Tovrstni modeli so

večinoma nastali v 70 in 80ih letih prejšnjega stoletja. Ocenili smo, da naša knjižnica pokriva večino teh modelov. Razmeroma kompletna zbirka teh modelov je podana v (Bowie et al., 1985). Razvoj računalnikov je omogočil prehod iz oddelčnih v prostorske sisteme, ki vsebujejo kompleksnejše matematični opise, t.j. navadne diferencialne enačbe preidejo v parcialne diferencialne enačbe. Reševanje parcialnih diferencialnih enačb je vezano na zahtevne numerične postopke, ki sami po sebi ne zagotavljajo natančne, oz. inženirsko sprejemljive rešitve. Zaenkrat so taki kompleksni prostorski sistemi skoraj nerešljivi z opisano metodo, predvsem zaradi prevelike računske zahtevnosti.

Glavne omejitve knjižnice znanja se torej nanašajo na fizikalno segmentacijo sistema, t.j. reševanje parcialnih enačb zaenkrat še ni mogoče. Prav tako ni mogoče modeliranje spremenljivega razmerja hranil v celicah primarnih in sekundarnih producentih.

6.3 Relevantnost in pravilvost odkritih modelov

Struktura rezultirajočih modelov je predvsem odvisna od samega eksperta (od specifikacije problema) tako, da tu ni vprašanja pravilne strukture odkritih modelov. Pač pa so diskutabilne tiste zadeve, ki se odkrijejo v postopku indukcije, t.j. optimizacije posameznih struktur na merjenih podatkih. Tudi sama narava modelov je taka, da z ustrezno optimizacijo parametrov lahko isti domeni vsilimo več struktur. V nizu dobljenih optimiziranih modelov za določeno domeno lahko najdemo več modelov različnih struktur, ki imajo minimalno razliko v napaki. To pomeni, da so v danem nizu rezultirajočih modelov tudi taki, ki niso povsem v skladu z ekspertnim poznavanjem domene. Vendar imajo ti modeli v večini primerov nekoliko večjo napako (MSE ali MDL) in so zatorej rangirani za 'ustreznimi' modeli. Seveda je ta napaka diskutabilna, kajti včasih simulacija 'neustreznega' in ustreznega modela pokaže zelo podobno, če ne celo enako prileganje k meritvam.

V naših primerih se je izkazalo, da LAGRAMGE odkrije modele, ki so v skladu z ekspertnim znanjem. Vzemimo za primer odkrivanje omejitvenih hranil za rast fitoplanktona. Na obeh nizih podatkov jezera Glumsø je LAGRAMGE odkril fosfor kot omejitveno hranilo, kar je v skladu z ekspertnim poznavanjem te domene. V nizih rezultirajočih modelov so bili tudi taki, ki so upoštevali raztopljeni dušik in fosfor za omejitveni hranili, a je bila njihova napaka večja. Za obmorske lagune je znano, da je dušik glavni omejitveni dejavnik za rast fitoplanktona (glej tudi poglavje 3.4). To se je potrdilo na podatkih Beneške lagune, kjer je pri vseh poskusih LAGRAMGE odkril dušik za omejitveno hranilo. Pri jezeru Kasumigaura lahko vsa hranila (fosfor, dušik in silicij) omejujejo rast totalnega fitoplanktona. Dušik kot pomembno hranilo so potrdile tudi druge raziskave (Bobbin in Recknagel, 2001), v kateri pa pri avtomatski gradnji modelov ni bil upoštevan silicij. LAGRAMGE je odkrival nekoliko različna hranila, glede na to, kako smo postavili eksperiment (priloga C), oz. kakšno ekspertno znanje smo vnesli v postopek odkrivanja modela. V enem primeru (brez upoštevanja vpliva zooplanktona) je bil vpliv vseh hranil zanemarljivo majhen za rast fitoplanktona, v drugem (upoštevan vpliv zooplanktona) pa so bila pomembna vsa hranila, t.j. silicij, dušik in fosfor, od

katerih ima fosfor zanemarljiv vpliv (priloga C). Torej imamo na Kasumigauri različne strukture modelov z podobno natančnostjo. Nekatere strukture uporabljajo silicij kot hranilo za rast fitoplanktona, druge pa ne. V takih primerih so verjetno najboljše dodatne meritve in ponovitev eksperimenta iskanja modela.

6.4 Diskusija rezultatov na realnih domenah

Običajno metode avtomatiziranega modeliranja potrebujejo veliko število podatkov, za indukcijo doberih (natančnih) modelov. Tu se je izkazalo, da je model odvisen od (1) znanja zajetega v knjižnici, (2) kvaliteta podatkov (Glumsø: veliko slabih podatkov) (3) kompleksnost ekosistema in (4) ekspertno znanje, ki ga vnesemo v postopek odkrivanja.

Zajeto ekspertno znanje v knjižnici je kjučnega pomena za odkrivanje konsistentnih modelov, ki sledijo osnovnim fizikalnim zakonitostim. To je potrdila primerjava dveh knjižnic iz obravnavane domene, t.j (1) enostavna knjižnica razvita v namen ilustracije orodja LAGRAMGE (Todorovski, 2003) in (2) razmeroma kompleksna knjižnica razvita v okviru te naloge (priloga A). Knjižnici smo preizkusili na podatkih jezera Glumsø (podatkovni niz A) in Beneške Lagune. Na primeru Beneške lagune smo s kompleksno knjižnico dobili strukture modelov, ki so konsistentnejše z domenskim znanjem, medtem ko je bila natančnost približno enaka v obeh primerih (priloga B). Na jezeru Glumsø sta obe knjižnici dali modele pravilnih struktur. Modela se razlikujeta v formulaciji posameznih procesov in v nekoliko natančnejši simulaciji modela, dobljenega s kompleksno knjižnico. Omeniti velja tudi prednosti enostavne knjižnice, ki se predvsem kažejo v računskih časih. Za enostavne probleme (modele) je nesmiselno uporabljati kompleksno knjižnico, kot se je to izkazalo na primeru jezera Glumsø (Tabela 15).

Kvaliteta podatkov vpliva na kakovst modelov. Jezero Glumsø je odličen primer za potrditev tega dejstva. Podatkovni niz A je vseboval samo 14 meritev, ki so jih eksperti interpolirali po lastni presoji. Interpolirane krivulje štirih ekspertov so predstavljale učne nize podatkov. LAGRAMGE je uspešno odkril modele fitoplanktona le na obdelanih podatkih prvega eksperta. Na drugem nizu podatkov (niz B) pa smo dobili model, ki ga je bilo mogoče tudi uspešno validirati na nevidenih podatkih. Na primeru jezera Bled smo sistem uspešno identificirali na podatkih iz leta 1996 z modelom treh enačb. Model opisuje dinamiko anorganskega fosforja, fitoplanktona in zooplanktona (*Daphnie hyaline*). Vendar sta simulaciji fosforja in fitoplanktona precej natančnejši, kot pa simulacija zooplanktona. Razloge lahko med drugimi iščemo v pomanjkljivih podatkih. Pretvorba zooplanktona iz števila individuumov v biomaso je bila zelo poenostavljena, pri čemer smo si pomagali s podatki iz literature (glej prilogo E). Zooplanktonovi plenilci (ribe) niso bili modelirani, ker ni bilo podatkov. Vprašljiva je tudi kvaliteta podatkov Beneške lagune. Za dve (od štirih) lokacij je LAGRAMGE uspešno odkril modele. Preostali dve lokaciji sta očitno vsebovali preveč šuma tako, da LAGRAMGE ni mogel odkriti modelov tudi s kompleksno knjižnico, čeprav gre za enako domeno. Po drugi strani pa je LAGRAMGE uspešno odkril nekaj modelov na jezeru Kasumigaura, čeprav smo imeli na razpolago le linearno interpolirane podatke. Vendar pa, tudi tu je bil

LAGRAMGE uspešnejši pri odkrivanju eno-letnih modelov. Slabšim modelom pri učenju na celotnem nizu je verjetno botroval prevelik šum v podatkih.

Vpliv kompleksnosti ekosistema. Znano je, da ekosistemi spreminjajo svojo strukturo s časom. Zato je težko odkriti nek (relativno) enostaven model, ki bi natančno simuliral situacijo čez daljše obdobje. V primeru Blejskega jezera smo najprej iskali model z učenjem iz celotnega niza podatkov, ki je sicer nakazoval delno ujemanje z meritvami a je bil kljub temu precej nenatančen (priloga E). Zato smo v naslednjem eksperimentu odkrili letne modele za skupno biomaso fitoplanktona. Za vsako leto je LAGRAMGE odkril modele, ki se med seboj razlikujejo tako po strukturi kot po vrednosti parametrov. Validacija modela odkritega na podatkih določenega leta na preostala leta je pokazala precejšnjo nenatančnost ujemanja z meritvami. Prav tako smo za leto 1996 odkrili model treh enačb, ki je natančno simuliral učne podatke, vendar pa je validacija na ostala leta pokazala precejšnje neujemanje. Nasprotno smo pri Kasumigauri uspeli najti reprezentativno leto. Model naučen na podatkih tega leta smo zadovoljivo validirali na podatkih preostalih let. Očitno ima jezero Kasumigaura ponavljajoče vzorce, ki jih je LAGRAMGE uspešno odkril.

Vpliv ekspertnega znanja. Z vnosom ekspertnega znanja v specifikaciji opazovanega sistema določamo vplive oz. procese, ki po našem mnenju vplivajo (ali bi lahko vplivali) na sistemske spremenljivke. Dodatni proces ali spremenjena definicija procesa lahko bistveno vpliva na samo kakovost in strukturo modelov. V primeru jezera Kasumigaura smo odkrivali dva tipa modelov za skupni fitoplankton. Prvi je izključeval vpliv zooplanktona, drugi pa vseboval še ta vpliv, t.j proces pašnje zooplanktona na fitoplanktonu. V prvem primeru je odkriti model pokazal, da imajo hranila zanemarljiv vpliv na rast fitoplanktona. Ko smo v postopek odkrivanja modela, vključili še proces pašnje pa se je izkazalo, da imajo nekatera hranila večji vpliv. Torej, se je struktura modela spremenila (priloga C). Pri Beneški Laguni se je zgodilo podobno pri odkrivanju modela za lokacijo 2. Ko smo v specifikaciji podali, da so omejitvena hranila za rast fitoplanktona lahko fosfor, nitrat ali amonij je LAGRAMGE vrnil model, ki se je slabše prilegal meritvam. Nato smo definicijo procesa rasti fitoplanktona popravili tako, da smo nitrat in amonij sešteli, ter za omejitvena hranila podali fosfor in skupni anorganski dušik. Tokrat je LAGRAMGE odkril model, ki se je bolje prilegal meritvam (priloga B).

7 Zaključki

V nalogi smo uporabili nov pristop k avtomatiziranemu modeliranju na področje ekološkega modeliranja jezer. Teza ima tri poglavitne prispevke (1) razvoj knjižnice znanja o modeliranju prehranjevalnih verig v jezeru za podporo avtomatskemu modeliranju, (2) uporaba knjižnize za gradnjo matematično (strukturno) pravilnih modelov in (3) aplikacija razvitega na realnih podatkih.

7.1 Knjižnica ekspertnega domenskega znanja

Razvili smo knjižnico znanja za ekološko modeliranje vodnih ekosistemov. Znanje je formalizirano v sintaksi procesnih razredov in obsega popis osnovnih/generičnih ekoloških procesov (kot so procesi evtrofikacije, npr. dotok hranil in njihovo kro ženje v sistemu, in populacijske dinamike, npr. rast, odmiranje, plenilstvo) v vodnih sistemih. Vsak procesni razred vsebuje tipične gradnike ekoloških modelov, ki ustrezajo posameznim procesnim razredom (npr. eksponentna ali logistična rast populacije). Knjižnica podpira modeliranje na osnovi masnih bilanc. Na podlagi tega, vsebuje tudi znanje o ustrezni kombinaciji procesnih razredov v skupni model sistema.

7.2 Evalvacija splošnosti predznanja, zajetega v knjižnici

Posplošeno znanje na osnovi procesov, oz. posameznih gradnikov za modeliranje le-teh, omogoča poenoten modularni pristop k gradnji modelov različnih vodnih ekosistemov. Z uporabo predznanja v knjižnici smo zapisali več znanih in uveljavljenih modelov vodnih ekosistemov. Iz knjižnice smo zgenerirali Vollenweider-jev model (Vollenweider, 1968), Imboden-ov model (Imboden, 1974) in model SALMO (Bendorf, 1979; Recknagel, 1980), ter tako pokazali splošnost domenskega znanja zajetega v knjižnici. SALMO spada med kompleksnejše modele tega tipa (box-modeli z navadnimi diferencialnimi enačbami). Generiranje tega modela je dokaz, da knjižnica vsebuje širok spekter znanja na tem področju in jo lahko uporabimo za modeliranje vodnih ekosistemov s kompleksnimi modeli.

7.3 Aplikacija na realnih primerih

Knjižnico smo evalvirali v kontekstu modeliranja realnih vodnih ekosistemov iz merjenih podatkov in domenskega predznanja, t.j. avtomatskega modeliranja s sistemom LAGRAMGE. Primeri vodnih ekosistemov, na katere smo aplicirali metodo so: Beneška laguna (priloga B), jezero Kasumigaura (priloga C), Blejsko jezero (priloga E) in jezero Glumsø (priloga B in D). Za vsako domeno smo zgradili uporabne modele in tako pokazali aplikabilnost metode na realnih podatkih. Vrednotenje modelov je pokazalo, da so zadovoljivo natančni in predvsem razumljivi ekspertom.

7.4 Ostali prispevki

- Vsi modeli odkriti na realnih domenah so pravilne strukture glede na ekspertno znanje, kar potrjuje konsistentnost zajetega znanja v knjižnici.
- Primerjava modelov dobljenih z uporabo knjižnice, razvite v okviru naloge, z modeli dobljenimi z uporabo enostavnejše knjižnice (Todorovski, 2003), je pokazala pomembnost in vpliv zajetega znanja v domenski knjižnici na rezultate, t.j. odkrite modele (priloga B).
- Na primerih smo pokazali strukturno dinamiko jezer. Jezera smo v različnih obdobjih identificirali z različnimi strukturami modelov.
- Pokazali smo pomembnost hranila silicij v skupni biomasi fitoplanktona. To se je izkazalo v primeru Kasumigaura in na primeru Blejskega jezera.
- Pokazali smo, da struktura modela, t.j. izbira procesov, ki bodo nastopali v modelou lahko bistveno vpliva na obnašanje modela. Velja tudi to, da imajo strukture modelov toliko prostostnih stopenj (pri umerjanju), da lahko isto domeno zadovoljivo natančno opišemo z več različnimi strukturami.
- Izčrpna in konsistentna baza podatkov za Blejsko jezero. Podatki meritev kakovostnih spremenljivk in pretokov so bili razdrobljeni po raznih inštitucijah, podvojeni ali izgubljeni, predvsem pa nepreverjeni. V namen uporabe merjenih podatkov za avtomatizirano ekološko modeliranje smo izdelali izčrpno in preverjeno podatkovno bazo za Blejsko jezero.

7.5 Nadaljnje delo

Zelo pomembna naloga v nadaljnjem delu bo usmerjena k približevanju izdelanega pristopa AM domenskim ekspertom. V ta namen je potrebno izdelati nek grafični vmesnik ki bo podpiral vsaj tri nivoje gradnje ekoloških modelov:

- Odkrivanje modelov z minimalnim vnosom ekspertnega znanja. Uporabnik poda naslednje podatke: fizikalna segmentacija jezera, spremenljivke stanja in tip spremenljivk stanja, ter neodvisne spremenljivke. Na osnovi teh podatkov LAGRAMGE lahko zgradi model z uporabo knjižnice (kjer so definirane masne bilance vsake sistemske spremenljivke) in meritev (odvisnih in neodvisnih spremenljivk), ki jih poda uporabnik.
- Odkrivanje modela z vnosom ekspertnega znanja o sistemu. Ta opcija zahteva od uporabnika definicijo strukture modela. Dodatno k podatkom, ki smo jih opisali zgoraj, uporabnik poda tudi povezave spremenljivk stanja s procesi in tudi definicijo procesov. Uporabnik torej sam definira masne bilance, glede na lastno poznavanje sistema. Procesi in njihove formulacije se izbirajo iz knjižnice znanja preko npr. grafičnega vmesnika. Območje parametrov in začetne vrednosti lahko določi uporabnik ali pa pusti predhodno definirane vrednosti.
- Tretja možnost bo ponujala implementacijo obstoječih modelov, takih kot smo jih opisali v poglavju 4. Uporabnik bo lahko apliciral modele take kot so, torej samo kalibracija parametrov, ali pa jih dodatno popravljal glede na svoje potrebe. Modifikacija modelov vključuje bodisi izbiro drugačnih formulacij posameznih procesnih razredov, ali pa razširjanje iskalnega prostora modelov s tem, da dopustimo več možnih formulacij za določen procesni razred.

Kar nekaj odprtih nalog je ostalo pri modeliranju Blejskega jezera. V okviru te naloge smo izdelali konsistentno bazo dosedaj merjenih podatkov. Zdaj je potrebna še natančnejša analiza podatkov, predvsem s strani ekspertov, ki se ukvarjajo s problematiko jezera. To bi bilo vodilo za izvedbo dodatnih meritev, na podlagi katerih bi dosledno identificirali prehranjevalne mreže v jezeru ter izdelali kompleksnejši konceptualni model. Model bi vključeval v prvem koraku fizikalno segmentacijo jezera na vsaj štiri dele, kar bi povzročilo veliko sistemskih spremenljivk, torej veliko kompleksnost modela. Odkrivanje takega modela zahteva zelo zahtevne računske vire, kar potegne za sabo naslednjo nalogo, t.j. zagotavljanje sodobnejših računalniških tehnologij, kot je recimo GRID tehnologija.

Prikladno bi bilo tudi avtomatizirati postopek učenja in validacije. Trenutno dobi uporabnik napake modelov le na učni množici podatkov. V bodoče naj bi dobili te rezultate tudi na predhodno specificirani testni množici in bi tako model vrednotili na obeh množicah.

Smiselno je izdelati knjižnice za ostale sorodne domene, kot so: modeliranje procesov v čistilnih napravah, modeliranje rek ipd. Nekatere domene zahtevajo reševanje parcialnih diferencialnih enačb, kar nalaga dodatno dodelavo LAGRAMGEa v tej smeri. Prav tako je nadaljnja evaluacija metode na realnih problemih nujno potrebna za dodatno potrditev metode.

8 Literatura

Andersen, T. (1997): Pelagic Nutrient Cycles: Herbivores As Sources in Sinks. Springer Verlag. Berlin.

Atanasova, N. in Kompare, B. (2002a): Modeling of waste water treatment plant with regression trees. Data Mining 2002. .

Atanasova, N. in Kompare, B. (2002b): Modeling of wastewater treatment plant with decision in regression trees. Workshop in Binding Environmental Sciences in Artificial Intelligence, ECAI. , pp. 6-9.

Atanasova, N. in Kompare, B. (2002c): Uporaba odlocitvenih dreves pri modeliranju cistilne naprave za odpadno vodo = The use of decision trees in the modelling of a waste water treatment plant. *Acta Hydrotechnica* **20**, 32.

Beck, M. (1983): A Procedure for Modeling, pp. 42. In Orlob, G. (Ed.): *Mathematical Modeling of Water Quality: Streams, Lakes in Reservoirs*, John Wiley & Sons, Chichester.

Belanche, L., Valdes, J., Comas, J., Roda, I. in Poch, M. (1999): Towards a Model of Input-Output Behaviour of Wastewater Treatment Plants using Soft Computing Techniques. *Environmental Modelling in Software* **14**, 409-419.

Bendorf, J. (1979): A contribution to the phosphorus loading concept. *Int. Revue ges. Hydrobiol.* **64**, 2, 177-188.

Bendoricchio, G., Coffaro, G. in De Marchi, C. (1994): A trophic model for Ulva rigida in the Lagoon of Venice. *Ecological Modelling* **75-76**, 485-496.

Bendoricchio, G., Coffaro, G. in Di Luzio, M. (1993): Modelling the photosynthetic efficiency for Ulva rigida growth. *Ecological Modelling* **67**, 2-4, 221-232.

Benz, J., Hoch, R. in Legovic, T. (2001): ECOBAS -- modelling in documentation. *Ecological Modelling* **138**, 1-3, 3-15.

Benz, J. in Hoch, R. (1997): Ein Modelldokumentationssystem. ASIM Simulationstechnik, 11. Symposium in Dortmund. , pp. 232.

Benz, J. in Knorrenschild, M. (1997): Call for a common model documentation etiquette. *Ecological Modelling* **97**, 1-2, 141-143.

Benz, J. in Voigt, K. (1996): Aufbau eines Systems zur strukturierten Suche von Informationsquellen für den Umweltschutz im Internet. Informatik für den Umweltschutz, 10. Symposium. , pp. 241.

Bertalanffy, L. (1972): The History in Starts of General System Theory. In Klir, G.J. (Ed.): *Trends in General Systems Theory*, John Wiley & Sons Inc, New York 047149190X.

Bierman, V. J., Dolan, D., Stoermer, J. G. in Smith, V. (1980): The development in Calibration of a Multi-Class Phytoplankton Model for Saginaw Bay, Lake Huron: *Great Lakes Environmental Planning Study*, Great Lakes Basin Comission, Ann Arbor, Michigan.

Bobbin in Recknagel (2001): Inducing explanatory rules for the prediction of algal blooms by genetic algorithms. *Environment International* **27**, 2-3, 237-242.

Bossel, H. (1994): *Modeling in Simulation*. A.K.Peters in Vieweg Verlag.

Bowie, G. L., Mills, W. B., Porcella, D. B., Campbell, C. L., Pagenkopf, J. R., Rupp, G. L., Johnson, K. M., Chan, P. W. H., Gherini, S. A. in Chamberlin, C. E. (1985): Rates Constants in Kinetic Formulations in Surface Water quality Modelling, US EPA, ORD, Athens, GA.

Breiman, L., Friedman, J., Olshen, R. in Stone, C. (1984): *Classification in regression Trees*. Wadsworth. Belmont.

Center for Water Research (CWR) (2003): DYRESM in CAEDYM, Center for Water Research, http://www2.cwr.uwa.edu.au/~ttfadmin/model/dyresmcaedym/.

Četina, M. (1988): Matematicno modeliranje dvodimenzionalnih turbulentnih tokov = Mathematical modelling of two-dimensional turbulent flows. *Acta Hydrotechnica* **6**, 5, 1-56.

Chapra, S. C. (1997): *Surface Water-Quality Modeling*. McGraw-Hill.

Chen, C. in Orlob, C. (1975): Ecological Simulation for Aquatic Environments, pp. 588. In Patten, B. (Ed.): *Systems Analysis in Simulation in Ecology*, Academic Press Inc., New York, New York.

Coffaro, G., Carrer, G. in Bendoricchio, G. (1993): Model for Ulva Rigida growth in the Lagoon of Venice: *UNESCO MURST Project: Venice Lagoon Ecosystem*, University of Padova, Padova, Italy.

Coffaro, G. in Bocci, M. (1997): Resources competition between Ulva rigida in Zostera marina: a quantitative approach applied to the Lagoon of Venice. *Ecological Modelling* **102**, 1, 81-95.

Coffaro, G. in Sfriso, A. (1997): Simulation model of Ulva rigida growth in shallow water of the Lagoon of Venice. *Ecological Modelling* **102**, 1, 55-66.

Comas, J., Džeroski, S., Gibert, K., Roda, I. in Sanchez-Marre, M. (2001): Knowledge discovery by means of inductive methods in wastewater treatment data. *AI Communications* **14**, 45-62.

Constanza, R. in Sklar, F. (1985): Articulation, accuracy in effectiveness of mathematical models: a review of freshwater in wetlin applications. *Ecological Modelling* **27**, 45-69.

DeAngelis, D. L. (1992): *Dynamics of Nutrient Cycling in Food Webs*. Chapmann & Hall. London.

Džeroski, S. in Todorovski, L. (1993): Discovering dynamics. Tenth International Conference on Machine Learning. , pp. 103.

Džeroski, S. in Todorovski, L. (1995): Discovering dynamics: From inductive logic programming to machine discovery. *Journal of Intelligent Information Systems* **4**, 89-108.

Džeroski, S. in Todorovski, L. (2003): Learning population dynamics models from data in domain knowledge. *Ecological Modelling* **170**, 2-3, 129-140.

Fasham, M. (1993): Modelling the marine biota, pp. 504. In Heimann, M. (Ed.): *The global carbon cycle*, Springer-Verlag, Berlin.

Fasham, M. (1995): Variations in the seasonal cycle of biological production in subarctic oceans. *Deep-sea Res. I* **42**, 1111-1149.

Frost, B. (1987): Grazing control of phytoplankton stock in the open subarctic Pacific Ocean: a model assesing the role of mesozooplankton, particularly the large kalanoidcopepods Neocalanus. *Mar. Ecol. Prog. Ser.* **39**, 49-68.

Gleick, J. (1991): *Kaos - Rojstvo nove znanosti*. DZS. Ljubljana.

Gruber, T.R. (1993): A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220.

Imberger, J. in Ivey, G. (1991): On The Nature Of Turbulence In A Stratified Fluid: 2. Application to lakes. *Journal Of Physical Oceanography* **21**, 5, 659-680.

Imboden, D. (1974): Phosphorus model of lake eutrophication. *Limnology in Oceanography* **19**, 297-304.

Imboden, D. (1979): Modelling of vertical temperature distribution in its implication on biological processes in lakes, pp. 561. In Jorgensen, S. (Ed.): *State of the Art in Ecological Modelling*, International Society of Ecological Modelling, Copenhagen.

Imboden, D. M. in Gachter, R. (1978): A dynamic lake model for trophic state prediction. *Ecological Modelling* **4**, 2-3, 77-98.

isee systems (2004): STELLA, isee systems, Lebanon.

Jørgensen, S. (2002): *Integration of Ecosystem Theories: A Pattern*. Kluwer Academic Publishers. Dordrecht, The Netherlins.

Jørgensen, S., Kamp-Nielsen, L., Christensen, T., Windolf-Nielsen, J. in Westergaard, B. (1986): Validation of a prognosis based upon a eutrophication model. *Ecological Modelling* **32**, 165-182.

Jørgensen, S. E. (1992): Parameters,ecological constraints in exergy. *Ecological Modelling* **62**, 163-170.

Jørgensen, S. E. in Bendoricchio, G. (2001): *Fundamentals of Ecological Modelling*. Elsevier Science Ltd. Amsterdam.

Kokar, M. (1986): Determining arguments of invariant functional descriptions. *Machine Learning* **4**, 403-422.

Kompare, B. (1995): The Use of Artificial Intelligence in Ecological Modelling: *Ljubljana, FGG; Royal Danish School of Pharmacy*, FGG, Ljubljana; Royal Danish School of Pharmacy, Copenhagen, Ljubljana, Copenhagen.

Krizman, V. (1998): Avtomatsko odkrivanje strukture modelov dinamicnih sistemov: *Fakulteta za racunalništvo in informatiko*, Univerza v Ljubljani, Ljubljana, Slovenija.

Langley, P., Simon, H., Bradshaw, G. and Zitkow, J. M. (1987): Scientific Discovery. MIT Press. Cambridge, MA.

Langley, P., Sanchez, J., Todorovski, L. in Džeroski, S. (2002): Inducing process models from continuous data. The Nineteenth International Conference on Machine Learning. .

Lotka, A. (1924): *Elements of Mathematical Biology*. William in Wilkins Co. Reprint: Dover Publications. New York, 1956.

Mahler, H. in Salomonsen, J. (1992): LAKE, Royal Danish School of Pharmacy, Copenhagen.

Malchow, H. (1994): Non-equilibrium structures in plankton dynamics. *Ecological Modelling* **75**, 123-234.

Monod, J. (1949): The Growth of Bacterial Culture. *Annual Review of Microbiology* **3**, 371-394.

Muetzelfeldt, R., Robertson, D., Bundy, A. in Uschold, M. (1989): The use of prolog for improving the rigour in accessibility of ecological modelling. *Ecological Modelling* **46**, 1-2, 9-34.

Nyholm, N. (1978): A simulation model for phytoplankton growth in nutrient cycling in eutrophic, shallow lakes. *Ecological Modelling* **4**, 2-3, 279-310.

O'Melia, C. (1972): An Approach to the Modelling of Lakes. *Schweiz. Z. Hydrol.* **34**, 1-34.

Orlob, G., Beck, M., Gromiec, M., Harleman, R., Jacquet, J., Jørgensen, S., Loucks, D., Mauersberger, P., Vasiliev, O. in Watanabe, M. (1983): *Mathematical Modelling of Water Quality: Streams, Lakes in Reservoirs*. John Wiley & Sons. Chichester.

Overbeck, J. (1989): Ecosystem Concepts, pp. 34: *Guidlines of Lake Management*, International Lake Environment Committee (ILEC).

Patten, B. in Jørgensen, S. (1995): *Complex Ecology: The Part-Whole Relation in Ecosystems*. Prentice Hall Ptr. New Jersey.

Quinlan, J. (1986): Induction of Decision Trees. *Machine Learning* **1**, 1, 81-106.

Quinlan, J. (1992): Learning with continuous classes. Proceedings AI'92 (Australian Conference on AI). , pp. 348.

Quinlan, J. (1993): Combining instance based in moodel based learning. 10th International Conference on Machine Learning. .

Rajar, R. in Četina, M. (1997): Hydrodynamic in water quality modelling. *Ecological Modelling* **101**, 195-207.

Recknagel, F. (1980): Systemtechnische Prozedur zur Modellierung und Simulation von Eutrophierungs-prozessen in stehenden und gestauten Gewässern: *Sektion Wasserwesen*, TU Dresden, Dresden.

Recknagel, F. (2004): Lake Kasumigaura data. In N. Atanasova (Ed.), Adelaide, AUS.

Reichart, P. (1998): AQUASIM 2.0 User Manual. Computer Program for the Identification in Simulation of Aquatic Systems, Swiss Federal Institute for Environmental Science in Technology (EAWAG), Dübendorf, Switzerlin.

Remec-Rekar, S. (1995): Življenska strategija in absorbcija fosforja pri nekaterih fitoplanktonskih vrstah Blejskega jezera-123, University of Ljubljana, Ljubljana.

Rismal, M. (1980): Presoja posameznih metod za sanacijo Blejskega jezera. *Gradbeni vestnik* **29**, 2-3, 34-46.

Rismal, M., Kompare, B. in Rajar, R. (1997): Contribution of hydrodynamic in limnological modelling to the sanitation of Lake Bled. Fourth International Conference on Water Pollution. , pp. 139.

Rizzoli, A. (2005): A Collection of Modelling and Simulation Resources on the Internet. http://www.idsia.ch/~andrea/simtools.html

Robertson, D., Bundy, A., Muetzelfeldt, R.,Haggith, M., Uschold, M. (1991): *Eco-Logic: Logic-based Approaches to Ecological Modelling*. MIT press. Massachusetts.

Roda, I., Comas, J., Sŕnches-Marré, M., Cortés, U., Lafuente, J. in Poch, M. (1999): Expert system development for a real wastewater treatment plant. Proc. of Chemical Industry in Environment III. , pp. 660.

Roesner, L., Giguerte, P. in Evenson, D. (1991): Computer program documentation for the Stream quality model QUAL-II, U.S. Evnironmental Protection Agency, Athens, Georgia.

Ross, A., Gurney, W. in Heath, M. (1994): A comparative study of the ecosystem dynamics of four fjords. *Limnol. Oceanogr.* **39**, 318-343.

Sanchez, M., Cortes, U., Bejar, J., DeGracia, J., Lafuente, J. in Poch, M. (1997): Concept Formation in WWTP by Means of Classification Techniques: A Compared Study. *Applied Intelligence* **7(2)**, 147-166.

Scavia, D. (1980): An Ecological Model of Lake Ontario. *Ecological Modelling* **8**, 49-78.

Simulistics (2005): SIMILE, Edinburgh Technology Transfer Centre, http://www.simulistics.com/index.htm, Edinburgh, Scotlin.

Sketelj, J. in Rejic, M. (1958): Preliminary account on the examination of Lake Bled. *Gradbeni vestnik* **61-64**.

Snodgrass, W. (1974): A Predictive Phosphorus Model for Lakes: Development in Testing, University of North Carolina, Chapel Hill, NC.

Steele, J. in Henderson, E. (1981): A simple plankton model. *Am. Nat.* **117**, 676-691.

Steinman, F., Banovec, P. in Šantl, S. (2001): Nacrtovanje razvoja vodovodnih sistemov z uporabo genetskih algoritmov = Genetic-algorithms-supported planning of water-supply systems. *Stroj. vestn.* **47; 6**, 263-279.

Steward, I. (1989): Does God Play Dice? The New Mathematics of Chaos. Basil Blackwell.

Streeter, H. in Phelps, E. (1925): A Study of the Pollution in the Natural Purification of the Ohio River: *Public Health Bulletin no. 146*, US Public Health Service, Colombus, Ohio.

Strmčnik, S. (1998): Širši konceptualni in metodološki okviri. In Strmčnik, S., Hanus, R., Juricic, Đ., Karba, R., Murray-Smith, D., Verbruggen, H., Zupancic, B. (Ed.): *Celostni pristop k racunalniškemu vodenju procesov*, FE in FRI, Ljubljana 961-6210-51-3.

Talbot, P., Thébault, J.-M., Dauta, A. in De la Noüe, J. (1991): A comparative study in mathematical modelling of temperature in light on growth of three microalgae potentially useful for wastewater treatment. *Water Research* **25**, 465-472.

Thebault, J.-M. in Salencon, M.-J. (1993): Simulation model of a mesotrophic reservoir (Lac de Pareloup, France): biological model. *Ecological Modelling* **65**, 1-2, 1-30.

Todorovski, L. (1993): Modeliranje dinamicnih sistemov z avtomatskim odkrivanjem zakonitosti: *Fakulteta za racunalništvo in informatiko*, Univerza v Ljubljani, Ljubljana.

Todorovski, L. (1998): Declarative bias in equation discovery: *Fakulteta za računalništvo in informatiko*, Univerza v Ljubljani, Ljubljana, Slovenija.

Todorovski, L. (2003): Using Domain Knowledge for Automated Modeling of Dynamic Systems with Equation Discovery: *Fakulteta zaračunalništvo in informatiko*, University of Ljubljana, Ljubljana, Slovenia.

Todorovski, L., Džeroski, S. in Kompare, B. (1998): Modelling in prediction of phytoplankton growth with equation discovery. *Ecological Modelling* **113**, 1-3, 71-81.

Todorovski, L. in Džeroski, S. (1997): Declarative bias in equation discovery. 14th International Conference on Machine Learning. , pp. 384.

Valiela, I. (1991): Ecology of Coastal Ecosystems, pp. 76. In Barnes, S.R. in Mann, H.K. (Eds): *Fundamentals of Aquatic Ecology*, Blackwell Science Ltd, Cambridge, GB 0632029838.

Verhulst, P.-F. (1845): Recherches mathématiques sur la loi d'accroissement de la population. *Nouv. mém. de l'Academie Royale des Sci. et Belles-Lettres de Bruxelles* **18**, 1-41.

Vollenweider, R. A. (1968): The scientific basis of lake in stream eutrophication with particular reference to phosphorus in nitrogen as eutrophication factors, Organisation for Economic Cooperation in Development, Paris.

Volterra, V. (1931): *Theorie mathematique de la lutte pour la vie*. Gauthier-Villars. Paris.

Washio, T. in Motoda, H. (1997): Discovering admissible models of complex systems based on scale-types in identity constraints. 15th International Joint Conference on Artificial Inteligence. , pp. 817.

Wei, B., Sugiura, N. in Maekawa, T. (2001): Use of artificial neural network in the prediction of algal blooms. *Water Research* **35**, 8, 2022-2028.

Witten, I. in Franck, E. (1999): Data Mining: Practical Machine Learning Tools in Techniques with Java Implementations (1st Edition ed.). Morgan Kaufman. San Francisco, CA.

Žagar, D., Rajar, R., Širca, A., Horvat, M. in Četina, M. (2001): Dolgotrajna 3D simulacija transporta in disperzije živega srebra v Tržaškem zalivu = Long-term 3D simulation of the transport in dispersion of mercury in the Gulf of Trieste. *Acta Hydrotechnica* **19**, 30, 25-43.

Zembovich, R. in Zytkow, J. M. (1992): Discovery of equations: Experimental evaluation of convergence. 10th National Conference on Artificial Inteligence, pp. 75.

Constructing a library of domain knowledge for automated modelling of aquatic ecosystems

Nataša Atanasova[1], Ljupčo Todorovski[2], Sašo Džeroski[2], Boris Kompare[1]

[1] *Faculty of Civil and geodetic Engineering, University of Ljubljana, Slovenia*
[2] *Jožef Štefan Institute, Slovenia.*

Abstract
Conceptual mathematical modelling of aquatic ecosystems comprises a considerable amount of knowledge reflected through a vast variety of different models that can be found in literature. While there is a growing interest in developing unifying documentation systems that allow storage of these models, not much work has been done yet on formalization and storage of the modelling knowledge itself. Such formalization would allow for better sharing and exchange of knowledge between experts on one hand and make it available to computational methods for modeling on the other. The knowledge library we develop here covers the knowledge in the domain of food web modelling in lakes based on differential equations. We illustrate the generality of the knowledge in the library through reconstruction of three well-known models of different complexity from the library, i.e., (Vollenweider, 1968), (Imboden, 1974) and SALMO model (Bendorf, 1979; Recknagel, 1980). We also illustrate how computational methods for model induction from data can benefit from the developed library of knowledge.

1. Introduction

There are at least three good reasons for modelling of aquatic systems, i.e. management, prediction and better understanding of the system. Mathematical conceptual models are mostly used and very popular among scientists (e.g., Jørgensen and Bendoricchio, 2001; DeAngelis, 1992; Chapra, 1997; and so on) due to their transparency and clearness to the domain experts. Many such models have been developed and published in literature - a comprehensive database of ecological models can be found on: http://dino.wiz.uni-kassel.de/ecobas.html (Benz and Voigt, 1996; Benz and Knorrenschild, 1997). The models in the database are documented under a unifying documentation system called ECOBAS (Benz and Hoch, 1997; Hoch et al., 1998; Benz et al., 2001). Despite their omnipresence, the task of establishing conceptual models is a very demanding task. The processes we are dealing with in ecological modelling are dynamic, interdependent, complex, and mostly not completely understood. The equations used for modelling are therefore adapted to (and reflecting) our incomplete knowledge. One consequence of this is that the quality of the obtained models greatly depends on the modeller skills and experiences and the other is that there is a variety of mathematical formulations for a specific ecological process. In other words there is no single suitable (corresponding, but not necessarily correct!) model for a specific system.

Computational methods for automated modelling (AM) aim at assisting scientists with the task of model building. A subclass of these methods, known as compositional modelling methods are used to build models by composing model fragments, commonly encoded in a library, into an adequate model of the entire

system. Main elements of the compositional modelling framework are the knowledge base, the specification of the observed system, and an algorithm capable of composing and evaluating different models. Rickel and Porter (1997) introduced and applied an automated modelling system TRIPEL, based on compositional modelling approach, for building models in the complex domain of plant physiology. In contrast to the compositional modelling approaches, machine learning methods are usually used to induce models from data only. Induction enables us to tackle various problems without the necessity to introduce any domain knowledge in the process of model construction. Successful applications of different machine learning (ML) techniques in ecology can be found for example in (Kompare, 1995; Kompare et al., 2001; Todorovski et al., 1998). The models induced by these methods are so called semi-transparent models, which mean that they can be partly explained and understood by an expert. However, the fact remains that they are induced from data, without incorporating any domain knowledge in the induction procedure.

Recently, a machine learning to AM has been developed, which make use of the domain expert knowledge to guide the process of induction towards models that follow the basic principles in the domain of interest (Džeroski and Todorovski, 2003; Todorovski, 2003). In the early days of the development of these tools (Todorovski and Džeroski, 1997) the knowledge had to be provided as an explicit specification of the space of candidate models. Now, these tools allow the user to provide higher-level (generic) domain knowledge about building mathematical models of complex real-world systems. Todorovski (2003) introduced a simple knowledge library for building models of aquatic ecosystems in order to confirm the applicability of knowledge-based induction to the task of modelling real-world aquatic ecosystems from noisy measurement data. However, the library is rather simple as it does not properly cover all aspects of knowledge from the domain of food web modelling in lakes.

The main focus of this paper is on building a comprehensive knowledge library that would cover most of the existing knowledge about modeling aquatic ecosystems. In order to be put in the library the knowledge needs to be coded according to the formalism (Todorovski, 2003) that is understood by the AM tool called Lagramge. The knowledge in the library comprises modeling of food web (or nutrients cycling) in a lake by following the mass conservation principle. It is formalized in terms of (1) taxonomy of variable types, (2) basic processes that govern the behavior of aquatic ecosystems, (3) alternative models of the basic processes, and (4) knowledge how to combine models of individual processes into a model of the entire ecosystem. Such formalization of the modeling knowledge paves a way towards easier sharing and exchange of knowledge between experts. It provides a solid unifying framework for both handcrafting ecological models as well as their automated induction from measured data.

The paper is organized as follows. First the domain knowledge about food web modeling in a lake, i.e., basic types of state variables, processes and mass balances is explained. Next the automated modeling framework is explained, starting with the library language formalism and how the library and the task specification are included in the model induction procedure, followed by simple example of model induction. In chapter four we evaluate the generality of the knowledge in the library

by deriving some well-known models from the library. Finally we give discussion, some guidelines for further work and conclusions.

2. Automated modelling framework

The procedure of automated modelling using the submitted, i.e., measured and suitably (re)interpreted data (Kompare, 1995) on the one side and the background knowledge on the other side is shown in Figure 1. The modelling knowledge is gathered and formalized in a library of domain-specific knowledge. The process of gathering and formalizing such knowledge for food web modelling is the topic of this paper. Next, modelling task has to be defined. This is done by user's specification of the observed system variables and processes that are expected to influence the behaviour of the system. Given a specification of modelling task at hand, Lagramge can now automatically transform the high-level knowledge from the library into an operational form of a grammar. This grammar now completely specifies the space of candidate models of the observed system. This is illustrated in the left-hand side of Figure 1.

Once we have the grammar, we can use equation discovery system Lagramge to heuristically search through the space of candidate models, match each of them against submitted data by fitting the values of the constant parameters. These models are evaluated (sorted) by two error measurements, i.e., mean square error (MSE) and minimum description length (MDL), are the output of Lagramge. Further details about the modelling framework from Figure 1 can be found in (Todorovski, 2003).

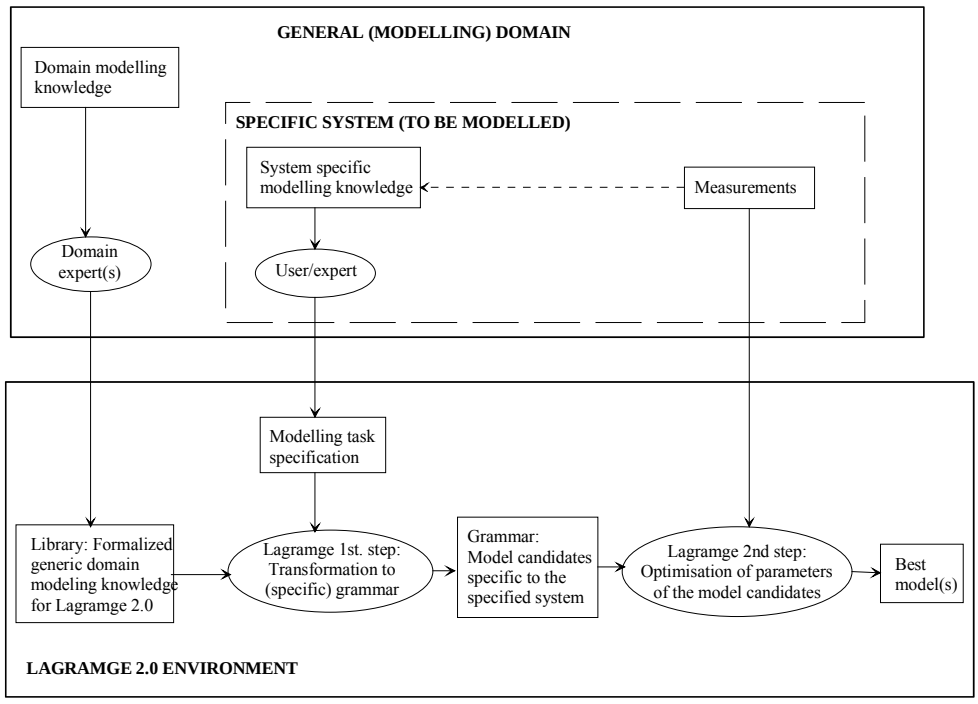


Figure 1: An automated modelling framework based on the integration of domain-specific modelling knowledge in the process of inducing models from data.

3. Conceptual mathematical modelling of aquatic ecosystems

3.1 Conceptualisation of the aquatic ecosystem

After defining the problem and modelling goals, the next step towards building a mathematical model is conceptualisation of aquatic ecosystem. It involves a choice regarding spatial segregation of the water body into a number of a discrete segments or layers as well as grouping and differentiation of biotic components according to their roles in the aquatic environment (Beck, 1983). To formulate the model of a system with a specific conceptualisation we need first to define the aquatic system. One of the most general definitions of a system was given by (Bertalanffy, 1972). He defines a system as a set of components interrelated between each other and with the environment. The components are connected with relations that usually represents exchange of matter, energy and information. Some components are related with the environment. The relations that have influence the components are called input to the system and those which have influence on the environment are output from the system.

Having this definition in mind we can conceptualize an aquatic system through the following groups of variables (Beck, 1983): (1) independent variables also called forcing or driving functions, or exogenous variables, representing the input in the system, (2) dependent variables, also called state or endogenous variables, which characterise the essential properties or behaviour of a system as functions of space and time. These variables represent the systems components, and (3) measured output variables (in most cases these are measurements of some of the state variables). Further development of this concept leads to a more detailed insight into the grouping of the biotic components or state variables and their relations. One way to observe the system is through the nutrient cycles in the system. State variables are connected with bio-chemical and physical processes, which are driven by the forcing functions. In this manner a process is a relation between the state variables and the independent variables (or forcing functions). In Figure 2 we show states and processes in one segment of a water body. The boxes represent the state variables, whereas the arrows stand for processes. The names of the physical and biochemical processes are given on the right hand side of the picture.

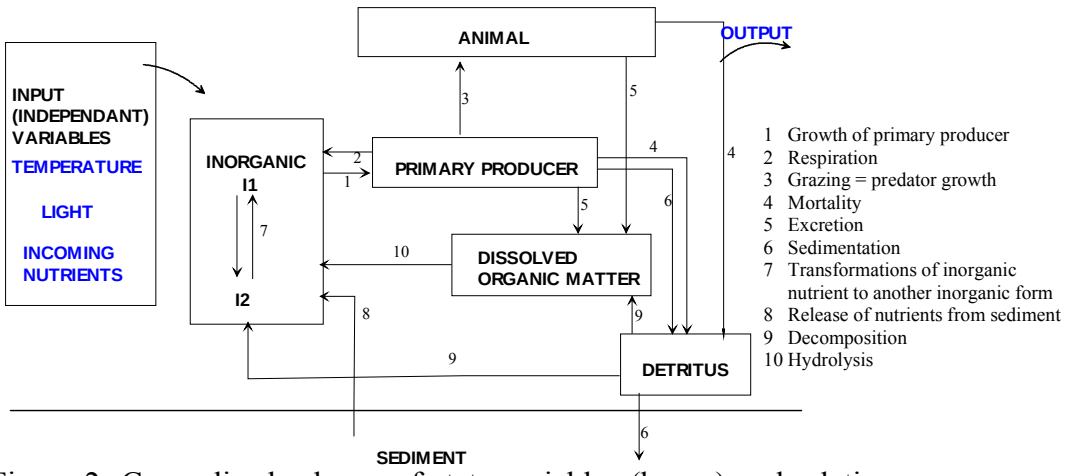


Figure 2: Generalized scheme of state variables (boxes) and relations or processes (arrows) in aquatic ecosystem

Most models of lake food webs distinguish between following types of state variables: dissolved inorganic nutrients (like inorganic nitrogen, phosphate, carbon), dissolved oxygen, primary producers (for example some species of algae), secondary producers (for example some species of zooplankton or fish), dissolved organic matter and detritus (boxes in figure 3). A type of state variable represents one or more state variables that are influenced similarly by the same type of processes. For instance two species of algae represent two state variables of same type (primary producer). Variables are related with physical, chemical and biological processes as shown in Figure 2.

3.2 From conceptual to mathematical model

Mathematically the conceptual model of the system can be represented in terms of ordinary differential equations (ODEs). State variables are time and space dependant, so these equations are commonly used to describe the temporal change of a specific state variable. The equation is set so that the mass conservation principle is fulfilled, i.e. all the processes that influence on the mass change of that variable are summarized in the equation. For example, the mass balance equation for primary producers includes following biological processes: growth of pp, non-predatory losses due to respiration, sedimentation and mortality, predatory losses due to grazing by zooplankton species and loss due to outflow (1):

$$\frac{dPP}{dt} = \text{growth} - \text{respiration} - \text{excretion} - \text{sedimentation} - \text{mortality} - \text{grazing} - \text{outflow} \quad (1)$$

Further, each of the processes in the mass balance equation is formulated with a mathematical expression. The expressions describe the relationship between the forcing functions and the state variables.

3.3 Alternative models for the basic processes

In general the processes can be divided into physical and bio-chemical. Physical processes include transport and mixing of matter, i.e. they connect the states between different compartments within the system, such as sediment-water interaction. To name few: settling of a substance, inflow (of nutrients), outflow of substances from the system, entrainment of a substance from one to another compartment and diffusion or mixing. Mathematical formulations of these processes are listed in Table 1.

Release of nutrients from sediment

The simplest approach to model this process is to specify an areal flux from the sediment in the mass balance equations for dissolved nutrients. This approach is used in QUAL II (Roesner et al., 1991; Duke and Masch, 1973; Johanson et al., 1980), and in SALMO (Recknagel, 1980). The other approach (e.g. Chen and Orlob, 1975; Smith, 1978; Thebault and Salencon, 1993) is to model nutrients in sediment as a separate dynamic pool (or layer).

Table 1: Physical processes in a lake: *C* stands for the observed substance (nutrient or biomass) concentration in [mass/volume], *A* is the area of the lake, *V* is the volume of the lake, *s* is the sedimentation rate in [1/time], *v* is the settling rate in [depth/time], and *h* is the water layer depth.

Process description	**expression**
Inflow of a substance (C_{in}) that contributes to the concentration of *C* in the lake	$\text{inflow} = C_{in}\frac{Q_{in}}{V}$
Load of a substance (C_{in}) from land that contributes to the concentration of *C*	$\text{land\_load} = C_{in}\frac{A_i}{V}$
Outflow of *C* caused by an outflow from the system	$\text{outflow} = C\frac{Q}{V}$
Settling of a substance with concentration *C*	$\text{settling} = s \cdot C \cdot \frac{A}{V}$
Diffusion of a substances from one layer (C_{i+1}) to another (C_i)	$\text{diffusion} = \frac{v_i}{h}(C_{i+1} - C_i)$

Bio-chemical or kinetic processes involve transformations of the nutrients from their inorganic form (like PO_4^{3-}, NH_3 or NO_3^-) through the food chain to organic form and their recycling back to inorganic form. These processes involved in nutrients' cycling are shown in Figure 2.

3.4 Chemical processes (first order kinetics):

Chemical processes include transformations of inorganic nutrients from one to another inorganic form, such as nitrification (transformation of ammonia to nitrate) and denitrification (transformation of nitrate to nitrogen), hydrolysis of the dissolved organic matter and decomposition of the particulate dead organic matter. In most cases they are described with first order kinetics equation (2).

$$\frac{dC}{dt} = k_{ref} \cdot f(T) \cdot C \tag{2}$$

where k_{ref} is the value of the rate coefficient at reference temperature (T_{ref}) in [1/time], *C* is the substance concentration in [mass/volume], *f(T)* a temperature adjustment function. Usual temperature adjustment formulation in these processes is the Arrhenius equation (3).

$$f(T) = \Theta^{T-Tref} \tag{3}$$

3.5 Primary producers' growth

In general the growth of primary producer growth can be stated as (4):

$$growth\_pp = \mu \cdot PP \tag{4}$$

where *PP* is a primary producer concentration in [mass/volume] and μ is the primary producer growth rate in [1/time].

There are three general formulations of the growth process according to the growth rate assumptions. Exponential model assumes a constant growth rate or unlimited growth. It supposes continuous reproduction, identical organisms, constant environment in space and time (e.g., resources are unlimited). Thus, the process is formulated as (4, where growth rate μ is constant. Unlike the exponential model, logistic model (Verhulst, 1845) suggests that the population growth is limited, i.e., it may depend on population density (5):

$$\mu = \mu_{\max} \cdot PP \cdot (1 - \frac{PP}{PP_{\max}}) \tag{5}$$

where μ_{max} is the maximal growth rate and PP_{max} is the upper limit of the primary producer concentration, referred to as carrying capacity. Population growth rate declines with the increase of PP, and reaches zero when $PP = PP_{max}$. If $PP > PP_{max}$ then the concentration of the primary producer declines.

Most ecological models account for growth, limited by several factors i.e., light, temperature and nutrients. The limiting factors or functions can be implemented differently in the expression for the growth rate. Most common formulation of the process is a product of the limiting functions of temperature, light and nutrients. The growth rate becomes (6):

$$\mu = \mu_{\max}(T_{ref}) \cdot f_1(T) \cdot f_2(L) \cdot f_3(N_1, N_2, N_3) \tag{6}$$

where $\mu_{max}(T_{ref})$ is the growth rate at the reference temperature (T_{ref}) and optimal conditions (in terms of food, temperature, and light), $f_1(T)$ is temperature adjustment of growth rate, $f_2(L)$ is the light limitation on growth rate, and $f_3(N_1,N_2,N_3)$ models the nutrients limitation on growth. N_1,N_2,N_3 are the limiting nutrients for the primary producer growth, such as phosphorus, carbon, and nitrogen.

Temperature influence on the growth rate can be modelled with three most common types of functions, i.e., linear, exponential, and optimal temperature adjustment functions. The linear (exponential) temperature function adjust the growth rate, so that it increases linearly (exponentially) with the temperature increase. The optimal function adjusts the growth rate coefficient, so that it increases up to some optimal temperature value, and when the temperature exceeds this value the growth rate decreases with the temperature. The expressions are listed in Table 2.

Table 2: Alternative temperature adjustment functions $f_1(T)$: T_{min} is minimal temperature, T_{ref} is the reference temperature, and T_{opt} is the optimal temperature.

Description	Expression	Reference
Linear temperature response functions above some minimal temperature T_{min}	$f(T) = \frac{T - T_{\min}}{T_{ref} - T_{\min}}$	
	For $T_{min} = 0$: $f(T) = \frac{T}{T_{ref}}$	This approach has been used e.g. in EXPLORE-I (Baca et al., 1973), in an early version of WASP (Di Torro et al., 1971)
Exponential temperature adjustment functions	$f(T) = \Theta^{T - Tref}$	Arrhenius or van't Hoff equation.
Optimal temperature adjustment functions	$f(T) = \exp\left(-2.3\left\lvert\frac{T - T_{opt}}{T_{opt} - T_{\min}}\right\rvert\right)$	Used in e.g. (Jørgensen, 1976), (Jørgensen et al., 1978)
	$f(T) = V^x e^{x(1-V)}$ $V = \frac{T_{\max} - T}{T_{\max} - T_{opt}} \quad x = \left[\frac{W(1+\sqrt{1+40/W}}{20}\right]^2$ $W = \ln(Q_{10})(T_{\max} - T_{opt})$ ** Simplified in the library: $f(T) = \left(\frac{T_{\max} - T}{T_{\max} - T_{opt}}\right)^x \cdot \exp\left[x\left(1 - \frac{T_{\max} - T}{T_{\max} - T_{opt}}\right)\right]$ where x is a parameter for estimation [1.7, 12]	(Shugart et al., 1974); implemented in CLEAN (Bloomfield et al., 1973), CLEANER (Scavia and Park, 1976), MS. CLEANER (Park et al., 1980).
	$f(T) = 2(1+b)\frac{x}{x^2 + 2bx + 1}$ $x = \frac{T - T_{\min}}{T_{opt} - T_{\min}}$	(Thebault and Salencon, 1993), implemented in the ASTER model.

** From the Arrhenius equation: $k(T) = k(T_{ref})\ \Theta^{(T - Tref)}$ or $Q_{10} = \Theta^{10} = k(T_x) / k(T_x - 10)$; $k(T) = k(T_{ref})\ Q_{10}^{(T - Tref)/10}$

The light influence is usually modeled with (1) saturation type of light limitation functions or (2) photoinhibition type of light functions. The first type of the light function can be expressed with a simple Monod expression (7), or with the Smith formulation (Smith, 1936) (8).

$$f_2(L) = \frac{L}{k_{sL} + L} \tag{7}$$

$$f_2(L) = \frac{a_1 L}{\sqrt{1 + (a_1 L)^2}} \tag{8}$$

where L is the intensity of the light useful for photosynthesis, K_{sL} is the half-saturation constant, and $1/a_1$ is the slope of the linear portion of the photosynthesis vs. light curve. Note that the unit of $a1$ is [1/light].

Photoinhibition type of light function takes into account that light can also have negative effect on primary producer growth (photosynthesis) when exceeding a certain value of light intensity, i.e. optimal light intensity. The function can be expressed with Steele formulation (Steele, 1965) (9). (Walker, 1975) used a slightly different expression (10).

$$f_2(L) = \frac{L}{L_{opt}} e^{\left(1 - \frac{L}{L_{opt}}\right)} \tag{9}$$

$$f_2(L) = \left(\frac{L}{L_{opt}}\right)^n e^{\left(1 - \left(\frac{L}{L_{opt}}\right)^n\right)} \tag{10}$$

where L_{opt} is the optimal light intensity for algal growth, and n adjusts the decline rate of the photosynthesis vs. light curve for light intensities above and below the optimum. The typical values for n are 0.67, 0.80, and 1.

(Talbot et al., 1991) proposed coupled effects of light and temperatures into one expression in their ASTER model:

$$\mu = \mu_{\max}(T_{ref}) g(T, L) f_3(N_1, N_2, N_3)$$

Nutrient limitation for primary producer growth, i.e., function f_3 is usually expressed with Monod expression. Limiting function for a single limiting nutrient is expressed as:

$$f_3(N) = \frac{N}{k + N} \tag{11}$$

Growth limitation by several nutrients is modelled by combining the single nutrient limitation functions. Different ways of combining have been proposed, e.g., Liebig's law of minimum (12), multiplication (13), or arithmetic mean (14).

$$f_3(N_1, N_2, N_3) = \min\left[f_3(N_1), f_3(N_2), f_3(N_3) \right] \tag{12}$$

$$f_3(N_1, N_2, N_3) = f_3(N_1) f_3(N_2) f_3(N_3) \tag{13}$$

$$f_3(N_1, N_2, N_3) = \frac{f_3(N_1) + f_3(N_2) + f_3(N_3)}{3} \tag{14}$$

Instead of Monod model we can also use the Monod^2 model (15) or the exponential model (16) as a single nutrient limitation function.

$$f_3(N) = \frac{N^2}{k + N^2} \tag{15}$$

$$f_3(N) = 1 - e^{-kN} \tag{16}$$

3.6 Secondary producers' producers

This process models predator–prey interactions, which include a predatory loss (due to e.g., grazing) of prey on one hand and the predator growth on the other. Note the important difference between primary and secondary producers' growth. While primary producers need all the inorganic nutrients to be present in the environment at the same time (and their concentration does not grow in absence of any), secondary producers can feed on any prey species (and their concentration grow even in the absence of others). In terms of mathematical models, this means that we can sum up concentrations of individual prey species (F_k) into a total food concentration (F_T) as in (17).

$$F_T = \sum_{k=1}^{n} F_k \tag{17}$$

In cases when predator prefers some prey species over others, we can take the selective feeding into account using weighted sum:

$$F_T = \sum_{k=1}^{n} pf_k \cdot F_k \tag{18}$$

where pf_k is food preference factor for F_k species.

Two types of the grazing process are usually used in literature. The first one uses the ingestion rate coefficient and the second uses the filtration rate, which states the amount of water filtrated per unit of zooplankton per time, since most of the zooplankton are filter feeders. Both formulations, i.e. using the grazing rate coefficient (19) and using the filtration rate (20) are shown below.

$$\text{feeding} = c_{g\max} \cdot f_1(T) \cdot f_3(F_T) \cdot SP \tag{19}$$

$$\text{feeding} = \sum_{k=1}^{n} F_k \cdot c_{f\max} \cdot f_1(T) \cdot f_3(F_T) \cdot SP \tag{20}$$

where c_{gmax} is the maximal zooplankton ingestion rate coefficient in [mass PP/(mass SP * time)], SP is the secondary producer concentration [mass/volume], c_{fmax} is the maximal filtration rate coefficient [volume/(mass SP * time)], and F_k is the individual prey concentration.

Both ingestion and filtration rate coefficients are temperature dependent and should be corrected with temperature adjustment functions $f_1(T)$, presented in the previous section. Food limitation function f_3 is formulated similarly as the nutrient limitation functions for primary producers' growth, equations (11), (15), and (16). Here we use the total available food concentration F_T instead of a single nutrient concentration (equation 21).

$$f_3(F_T) = \frac{F_T}{k + F_T} \tag{21}$$

3.7 Respiration and excretion

Excretion by primary producers and animals contribute significantly to nutrient recycling. Loss of an organism due to respiration or excretion can be generally written as a first order equation (see equation (1), where the rate coefficient is a function of temperature and/or physiological conditions of the organism. Many models include only temperature influence. (Scavia, 1980) and (Recknagel, 1980) in his SALMO model for example use a model that relates the algae respiration rate to the physiological conditions of the algal cells. The expression represents a sum of two components: a low maintenance rate representing periods of minimal growth and a rate which is proportional to the maximum growth rate limited by the growth limitation factors. Formulations for primary producer's respiration rate are listed in Table 3.

In case of zooplankton (Table 4) respiration rate is mostly modelled with first order kinetics, (see equation (1), i.e., only temperature influenced. Some models divide the respiration into 1) basal metabolism and digestion energetics and 2) active respiration rate, i.e. the additional respiration associated with zooplankton activity. This approach is used by e.g. (Scavia, 1980) or (Recknagel, 1980).

Table 3: Primary producer respiration rates (r):r_{ref} is respiration rate at reference temperature, r_{opt} is respiration rate at optimal temperature, k_r is maximum incremental increase in respiration under conditions of maximal growth, r_{min} is respiration at 0°C r_1 is a portion of gross photosynthesis rate which is consumed by respiration additionaly to the basis respiration

Expression	Description
r=const	exponential model
$r = r_{ref} \cdot f(T)$	rate influenced by temperature, commonly used in e. models
$r = r_{ref} \cdot f(T) + k_r \cdot f(T) \cdot f(N) \cdot f(L)$	(Scavia, 1980)
$r = (r_{opt} - r_{\min}) \cdot \frac{T}{T_{opt}} + r_{\min}) + r_1 \cdot \mu$	used in SALMO by (Bendorf, 1979) and (Recknagel, 1980)

Table 4: Secondary producer respiration rates (r): r_{ref} is respiration rate at reference tempeerature, r_{min} is minimum endogenous respiration under starvation conditions at reference temperature, r_a is active respiration rate, kr is fraction of ingested food which is respired, ropt is respiration rate at optimal temperature for feeding and maximum ingestion rate, rmin is respiration rate at 0°C and optimal food suply.

Expression	Description, references
r=const	exponential model
$r = r_{ref} \cdot f_1(T)$	rate influenced by temperature, commonly used in e. models
$r = r_{\min} \cdot f_1(T) + r_a \cdot f_1(T) \cdot f_3(F_T)$	(Scavia et al., 1976), (Scavia, 1980)
$r = r_{ref} \cdot f_1(T) + k_r \cdot f_1(T) \cdot C_g$	(Scavia et al., 1976), (Park et al., 1980)
$r = (\frac{r_{opt} - r_{\min}}{C_{g\max}} \cdot C_g + r_{\min}) \cdot \frac{1}{r_{opt}} \cdot ((r_{opt} - r_{\min}) \cdot \frac{T}{T_{opt}} + r_{\min})$	used in SALMO by (Bendorf, 1979) and (Recknagel, 1980)

3.8 Mortality

Nonpredatory mortality of primary producers includes processes like senescence, bacterial decomposition of cells (parasitism) stress-induced mortality, due to severe nutrient deficiencies, extreme environmental conditions, or toxic substances. Commonly, this process is modelled when no other loss process, such as settling is included. Mortality rate can be formulated as a constant or a temperature adjusted rate. Some models relate the mortality rate to the physiological conditions of the algal cells. For example (Scavia and Park, 1976) use the growth limitation factor as a measure of cell health. When the limiting factor is near the value of 1 (low limitation) then the mortality is low and vice versa. (Nyholm, 1978) uses a Monod saturation function of the algal concentrations. Primary producers' mortality rates are shown in Table 5.

Table 5: Primary producer mortality rates (m): m_{ref} is mortality rate at reference temperature

Expression	Description and references
m=const	exponential, non-limited model
$m = m_{ref} \cdot f_1(T)$	temperature influenced model
$m = m_{ref} \cdot PP \cdot f_1(T)$	second order kinetics, temperature influenced
$m = m_{ref} \cdot f_1(T) \cdot (1 - f_2(N) \cdot f_3(L))$	(Scavia and Park, 1976)
$m = m_{ref} \cdot f_1(T) \cdot \frac{PP}{k + PP}$	(Nyholm, 1978)

Zooplankton mortality is modelled with simple temperature dependant expression, when zooplankton predators are modelled separately. But if zooplankton mortality represents the closure term in the food-chain model then more complex expressions are used, which also account for the predator influence. (Bierman et al., 1980) for example use a second order formulation when the zooplankton density exceeds a certain level. Other formulations include quadratic (e.g. Steele and Henderson, 1981; Fasham, 1995), hyperbolic (e.g. Frost, 1987; Fasham, 1993; Ross et al., 1994) and sigmoid (Malchow, 1994) closure term. See Table 6 for mortality rate formulations.

Table 6: Secondary producer mortality rates (m): m_{ref} is mortality rate at reference temperature, m_1 is mortality rate below the critical animal density at reference temperature, k_m is density dependant mortality coefficient for increased mortality above the critical animal density, k is half saturation constant, m_{min} is minimal mortality rate and m_{opt} is mortality rate at optimal temperature

Expression	Description and references
m=const	exponential, non-limited model
$m = m_{ref} \cdot f_1(T)$	temperature influenced model
$m = m_{ref} \cdot f_1(T) \cdot SP$	quadratic, e.g. (Steele and Henderson, 1981), (Fasham, 1995)
$m = m_1 \cdot f_1(T) + SP \cdot k_m \cdot f_1(T)$	(Bierman et al., 1980)
$m = m_{ref} \cdot \frac{SP}{k + SP}$	hyperbolic expression; (Frost, 1987); (Fasham, 1993); (Ross et al., 1994)
$m = m_{ref} \cdot \frac{SP^2 \cdot}{k^2 + SP^2}$	Sigmoid expression; (Malchow, 1994)
$m = m_{\min} + m_{opt} \cdot T \frac{SP}{k + SP}$	SALMO model; (Bendorf, 1979) and (Recknagel, 1980)

3.9 The oxygen model

A simple oxygen model is included in the knowledge base. The general mass balance for oxygen can be written as (22):

$$\frac{dO}{dt} = reaeration - consumptions + production \tag{22}$$

Reaeration represents the exchange of oxygen between air and water. Usually it is modelled as (23):

$$reaeration = k_L \cdot (C_S - C) \tag{23}$$

where C is concentration of dissolved oxygen, Cs is oxygen concentration at saturation and k_L is the transfer coefficient [1/time].

Consumption of oxygen in lakes is due to following processes: microbial degradation of dissolved and particulate organic matter (decomposition), oxidation of nitrogen (nitrification), sediment oxygen demand (SOD) including oxidation of settled organic matter and respiration of benthic biota. Oxygen consumption processes, i.e. nitrification, decomposition and respiration are expressed in same terms as explained above. The processes are multiplied by a stoichiometric factor that converts the specific mass into oxygen units. For example oxygen consumption due to microbial degradation of organic matter (decomposition) is usually modelled by first order reaction (24).

$$cons\_decomposition = -k_D \cdot D \cdot Y_{O:C} \tag{24}$$

where D is organic matter concentration, k is mineralization coefficient and Y is stoichiometric coefficient of transformation from org. C to O [mgO2/mgC]

Finally, oxygen production in photosynthesis is expressed in terms of growth of primary producers:

$$ox\_prod = \mu \cdot PP \cdot Y_{O:PP} \tag{25}$$

where m is the growth rate of primary producer [1/time] PP is primary producer concentration [mass/volume] and $Y_{O:PP}$ is oxygen production per unit of mass PP [mass oxygen/mass PP].

4 Encoding the lake modelling knowledge into a knowledge library

The main goal of the AM approach used in this research is to use the background expert knowledge in the procedure of automated model induction from data. In order to be used in the automated modelling procedure the modelling knowledge needs to be coded in appropriate formalism (Todorovski, 2003) understandable to the AM tool (Lagramge) and stored in a domain specific library. The knowledge library is

written in a form of generic processes connected in combining schemes, which in ecological language represent a mass balances for the state variables. Thus, the library offers knowledge for modelling of the aquatic system processes through mass balances of the system‘s state variables. Figure 2 shows the basic processes in the knowledge library and their influence on the state variables.

The knowledge in the library is coded in form of (1) taxonomy of variable types, (2) taxonomy of basic process classes that govern the behavior of aquatic ecosystems, (3) alternatives for modeling processes from each class, and (4) knowledge how to combine models of individual processes into a model of the entire ecosystem (Todorovski, 2003).

The types of variables defined in the library correspond to the types presented in Section 3. Taxonomy of variable types is given in Table 7, which is schematically represented in Figure 3.

Table 7: Taxonomy of variables in the knowledge library for lake modelling

Variable type	Description	dependant (state)/independent
type Concentration **is real**	concentration of a substance	generic
type Light **is real**	light intensity	independent
type Temperature **is real**	temperature	independent
type Precipitation **is real**	precipitations	independent
type Flow **is real**	flow rate	independent
type Area **is real**	contributing area of the incoming nutrients	independent
type Inorganic **is Concentration**	dissolved inorganic nutrients	dependant
type Population **is Concentration**	concentration of a population	generic
type Detritus **is Population**	particulate dead organic matter	dependant
type Oxygen **is Concentration**	dissolved oxygen	dependent
type Dom **is Concentration**	dissolved organic matter	dependant
type Primary_producer **is Population**	primary producers	dependant
type Animal **is Population**	secondary producers	dependant

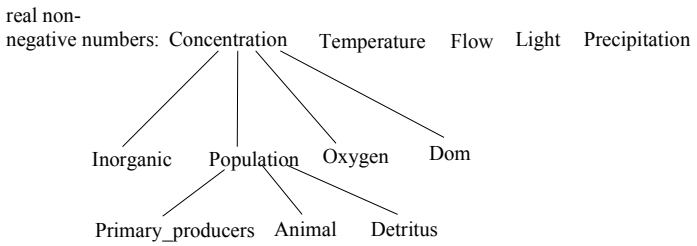


Figure 3: Hierarchical representation of variable types and sub-types in the knowledge library for lake modelling.

Concentration is generic variable type that is defined as a non-negative real number. It has four subtypes, i.e. Inorganic, representing the dissolved inorganic nutrients, Population representing the organic particulate matter, Dom, denoting a dissolved organic matter and Oxygen representing dissolved oxygen concentration. Population has again three sub-types – Primary_producer, Animal, and Detritus. Note that the

hierarchical representation defines inheritance of types, i.e., a variable of a type Animal also belongs to Population and Concentration types. To model interactions between many species, e.g., when we model interaction between a single primary producer and more than one inorganic nutrient, the later is specified as a set of variables. Thus, a variable type can be defined that denote a set of variables of the same type. Declaration of a set of primary producers is given below:

type Inorganics is set(Inorganic)

Furthermore, we have to specify the basic bio-chemical and physical process classes that influence the state variables in the domain of interest. Figure 2 specifies these process classes – each process class correspond to one of the enumerated arrows there. Each process class definition specifies the types of the variables involved in (or influenced by) the process, and typical alternative expressions used to model processes in the class. We define as many process subclasses of the process class as there are alternative models for that class. Each model is specified as an expression template that includes variables involved in the process class and generic constant parameters, which are specified with the symbol *const(name, lower_bound, initial_value, upper_bound)*. The values of the generic parameter constants are to be fitted against measurement data in the model calibration phase. Expressions may also refer to other processes as well as functions, which are defined separately in the library. The definition of function class is equivalent to the process class definition except the keyword *process class* is replaced with the keyword *function class*. The difference is in the fact that functions do not necessary represent processes in the domain of interest.

For example, consider the definition of the primary producer growth process class, presented in Figure 4. From this illustration we can understand the tree structure of the formalism. The primary producer (*pp*) growth can be influenced by a set of inorganic nutrients (*ns*), temperature (*ts*), and light (*ls*). Recall from Section 3, that the growth is modelled using one of the alternatives presented in equations (4, (5, and (6, which correspond to three different models of *Growth_rate*, which is multiplied by concentration *pp*. When the third (limited growth) alternative is being used, we have to consider the modelling alternatives for food limitation (3 different forms), light limitation (two forms), and temperature influence (three forms) as specified in Section 3. The formal encoding of the process class can be found in Appendix 1.

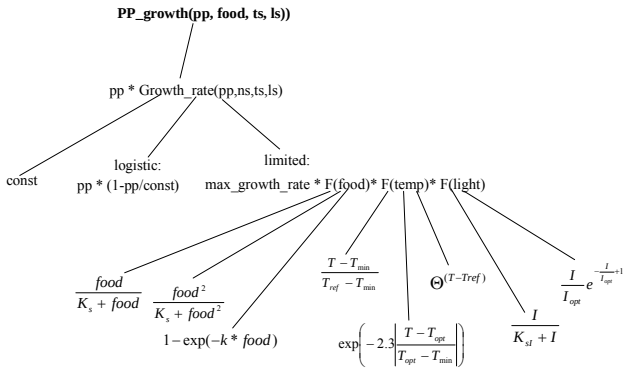


Figure 4: Encoding the modelling knowledge for the process class of primary producer growth.

Finally, in order to build a model of the whole system, we have to know how to combine the models of individual process classes together. Mass balance equation ((1) gives a recipe for combining expressions used to model different process classes into a differential equation for the temporal change of primary producer concentration. In our formalism, we have to specify such a combining scheme for each system variable type.

Consider the two combining schemes presented in Table 8. The first one is for inorganic nutrients while the second is for primary producers. [1] The temporal change of primary producer is influenced by eight process classes. Two process classes positively influence the primary producer change: *PP_growth* and *Diffusion*. The latter refers to processes of interchange of primary producer concentrations between lake layers – the expression *sum({pp1}, true, Diffusion(pp, pp1))* denotes that we want to sum up all the processes that contribute to the *pp* concentration in the current layer from all the neighboring layers (*pp1* denotes primary producer concentration in each of the neighboring layers). Note however, that the *Diffusion(pp1, pp)* denotes the concentration flow in the opposite direction (from current to neighboring layers) – that is why the influence of these processes to the *pp* concentration is negative. Similarly, the process class of *PP_growth* positively influences the concentration of primary producer (second combining scheme), while it causes decrease of the concentrations of inorganic nutrients consumed by *pp* (first combining scheme). The condition *i in food* specifies that only those inorganic nutrients consumed by *pp* are negatively influenced by its growth. While the *sum* aggregation function is usually used to bring together all the processes from a particular class that influence the observed system variable, it also denotes that in absence of such processes, the value of the corresponding term is zero.

The complete knowledge library for modelling lake ecosystems is given in Appendix 1.

4.1 Modelling task specification- using the library for inducing different model structures

The presented library encodes knowledge that can be used for modelling an arbitrary lake ecosystem. User faced with a particular modelling task has to specify a list of observed variables and processes in the observed system (see also **Figure 1**). Observed variables are declared in following way:

variable variable_type ‘variable_name’

We can use the word **system** in front of the word **variable** if the variable is state variable and we want to discover an equation for that specific variable. The processes are declared by the word *process*, process name and the process arguments. For example the process:

process PP_growth(phyto, {ortp}, {temp}, {light}) **phyto_limited_growth**

[1] Note that for clarity reasons, the combining scheme for inorganic nutrients is only partially presented.

describes that the phytoplankton (*phyto*) growth is limited by temperature, light, and inorganic nutrient *ortp*. Note that the variables in the curly brackets, i.e. {ortp}, {light} and {temp} denote sets. They can include arbitrary number of elements – note that empty sets are also allowed as in:

process PP_growth(phyto, {phosp, nitro}, {temp}, {})
phyto_limited_growth_no_light

which denotes that phytoplankton growth is limited by two inorganic nutrients (*phosp* and *nitro*) and temperature and is not influenced by the light intensity.

Table 8: A segment of the combining schemes specification for the variable types Inorganic and Primary producer.

```
combining scheme Lake(Inorganic i)
        time_deriv(i) =
         ...
                - sum({pp, food, ts, ls}, i in food, const(conv, 0.0005,0.002, 0.009)*PP_growth(pp,
food, ts, ls))

combining scheme Lake(Primary_producer pp)
        time_deriv(pp) =
                +sum({food, ts, ls}, true, PP_growth(pp, food, ts, ls))
                - sum({ns,ts,ls}, true, Respiration_PP(pp,ns,ts,ls))
                - sum({ns,ts,ls}, true, Mortality_PP(pp, ns,ts,ls))
                - sum({}, true, Outflow(pp))
                - sum({ts}, true, Sedimentation(pp,ts))
                +sum({pp1}, true, Diffusion(pp,pp1))
                - sum({pp1}, true, Diffusion(pp1,pp))
                - sum({a, food, ts}, pp in food, Feeds_on(a, food, ts))*Food_pref(pp)
```

5. Generality of the domain knowledge in the library - deriving well-known models

The generality of the knowledge comprised in the library was evaluated by generating grammars for several well-known ecological models of different complexity. One of the first ecological models for estimation of a lake trophic state was Vollenweider's model (Vollenweider, 1968). The model consists of a single equation (26) that describes the change in phosphorus concentration in a lake. Thus, it has one state variable ([P]) and three processes, i.e. inflow of phosphorus to the lake, loss of phosphorus due to sedimentation and outflow of phosphorus from the lake.

$$\frac{d[P]}{dt} = \frac{P_{in}}{V} - \frac{K_{sed}}{h} \cdot [P] - \frac{Q}{V} \cdot [P] \qquad (26)$$

Task specification to describe this model is given in Table 9. Variables are declared in the first four lines, i.e. one state variable, i.e. phosphorus (*p* of type Inorganic) and three independent variables, i.e, phosphorus concentration in the inflow (*p_in* of type Inorganic), inflow and outflow flow rates (*q_in* and *q_out* of type Flow). In the

next lines processes in the system are given. The process *Inflow* represents inflow of phosphorus (*p_in*) with the flow rate *q_in* to the system, contributing to the concentration of phosphorus in the lake (*p*). Next process, i.e. *Outflow* represents the outflow of *p* with the flow rate *q_out*. Finally, the process *Sedimentation* is declared with a single attribute *p* representing the sedimentation of the state variable *p*.

Table 9: Task specification for the Vollenweider's model

variable Inorganic p_in system variable Inorganic p variable Flow q_in variable Flow q_out	process Inflow(p, p_in, q_in) inflow process Outflow(p, q_out) outflow process class Sedimentation (p) sedimentation

Given the task specification from Table 9 and the knowledge library Lagramge transformed the task into a grammar of possible model structures. The grammar specifies the candidate models in following way. First the combining schemes (mass balances) for the system variables (p) are applied:

p' = Inflow(p, p_in1, q_in) – Outflow(p, q_out) – Sedimentation(p)

Each combining scheme (only one in this case) adequately combines the processes that influence on the specific state variable. The automated modelling system then enquires the taxonomy for all possible expressions used to model each of the component processes (that is Inflow, Outflow, and Sedimentation, see Table 10).

Table 10: The grammar of model structures for the Vollenweider's model task specification (Table 9)

```
Outflow(p, q_out) -> p * q_out / const[volume];
Inflow(p, p_in1, q_in) -> p_in1 * q_in / const[volume];
Sedimentation(p) -> p * const[s:0.0001:0.02:0.3] / const[h:10:10:10];
```

From Table 10, it is evident that for this task we have a single model structure, identical to the Vollenweider's model (26).

Further development of the lake models included more complexity, i.e. modelling more state variables and processes. Imboden (1974) suggested a two-compartment (stratified) model for soluble reactive (inorganic) phosphorus (SRP) and phosphorus in algae or particulate reactive phosphorus (PRP), or phosphorus in phytoplankton. The model includes four state variables (SRP in epi- and hypolimnion and PRP in epi- and hypolimnion) and 14 processes as shown in Figure 5. In its first version of the Imboden model the processes were formulated with first order equations. The model was improved (Imboden and Gachter, 1978) by replacing first order kinetics with the Monod one. All variations of the original model can be generated from the knowledge library with the task given in Table 11. This is evident from the grammar (Table 12) obtained from the knowledge library and the given task specification.

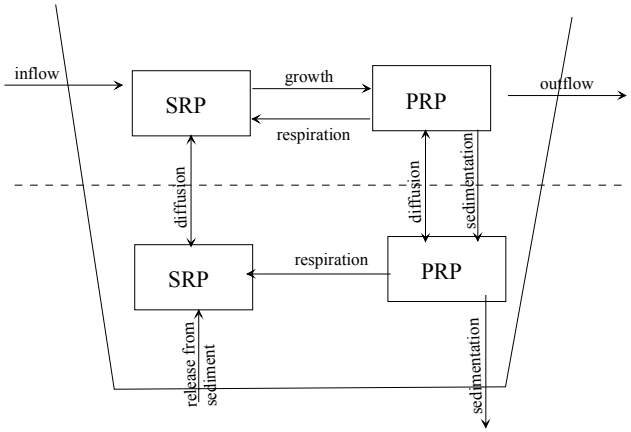


Figure 5: Conceptual model for stratified lake (Imboden, 1974)

Table 11: Task specification for Imboden's model

variable Inorganic ortp_in	process PP_growth(phyto_e, {ortp_e}, {}, {}) gr1
variable Flow q_in	process Respiration_PP(phyto_e, {},{},{}) respiration1
variable Flow q_out	process Respiration_PP(phyto_h, {},{},{}) respiration2
system variable Inorganic ortp_e	process Respiration_PP_nut(ortp_e,phyto_e,{},{},{}) respiration_nut1
system variable Inorganic ortp_h	process Respiration_PP_nut(ortp_h,phyto_h,{},{},{}) respiration_nut2
system variable Primary_producer phyto_e	process Sedimentation_to(phyto_e,phyto_h) sed1
system variable Primary_producer phyto_h	process Sedimentation_to(ortp_e,ortp_h) sed2
process Inflow(ortp_e, ortp_in,q_in) inflow1	process Sedimentation(ortp_h) sed3
process Outflow(ortp_e, q_out) outflow1	process Sedimentation(phyto_h) sed4
process Diffusion(ortp_e,ortp_h) diff1	process Sediment_release(ortp_h,{}) sed_release1
process Diffusion(phyto_e,phyto_h) diff3	

First, the mass balance equations (combining schemes of processes) are listed for the state variables: ortp_e, ortp_h, phyto_e and phyto_h (lines 1, 2, 3 and 4 in Table 12). Note that processes are the same as those given in Figure 5. Further in the grammar we show the variations of the primary producer growth process for phytoplankton in epilimnion (line 5 in Table 12), in this case unlimited growth, which represents a first order kinetics formulation (lines 6 and 8 in Table 12 and limited growth representing nutrient limited growth using the Monod formulation (see lines 7, 9, 12 and 13 in Table 12). Temperature and light are not taken into account by this model. The terms for those influences are equal to 1, which means no influence (lines 10 and 11 in Table 12).

Table 12: A segment of the Imboden's model grammar

1. ortp_e' -> Inflow(ortp_e, ortp_in, q_in) – Outflow(ortp_e, q_out) + Diffusion(ortp_e, ortp_h) + Respiration_PP_nut(ortp_e, phyto_e) - Sedimentation_to(ortp_e, ortp_h) - const[conversion_factor:0.0005:0.002:0.009] * PP_growth(phyto_e, ortp_e)

2. ortp_h' -> - Diffusion(ortp_e, ortp_h) + Respiration_PP_nut(ortp_h, phyto_h) + Sediment_release(ortp_h) – Sedimentation(ortp_h) + Sedimentation_to(ortp_e, ortp_h)

3. phyto_e' -> +PP_growth(phyto_e, ortp_e) - Respiration_PP(phyto_e) - Sedimentation_to(phyto_e, phyto_h) + Diffusion(phyto_e, phyto_h)

4. phyto_h -> - Respiration_PP(phyto_h) – Sedimentation(phyto_h) + Sedimentation_to(phyto_e, phyto_h) – Diffusion(phyto_e, phyto_h);

…

5. PP_growth(phyto_e, ortp_e) -> variable_phyto_e * Growth_rate(phyto_e, ortp_e);
6. Growth_rate(phyto_e) -> PP_growth_unlimited(phyto_e);
7. Growth_rate(phyto_e) -> PP_growth_limited(phyto_e);

8. PP_growth_unlimited(phyto_e) -> const[growth_rate:0.05:0.4:3];
9. PP_growth_limited(phyto_e) -> const[max_growth_rate:0.05:0.4:3] * Food_limitation(ortp_e)* Temperature_influences() * Light_limitations();

10. Light_limitations() -> const[_:1:1:1];
11. Temperature_influences() -> const[_:1:1:1];
12. Food_limitation(ortp_e) -> Food_limitation_type_1(ortp_e);
13. Food_limitation_type_1(ortp_e) -> ortp_e / (ortp_e + const[sat_rate:0.0005:0.02:0.05]);

The last example of model generation is a fairly complex model SALMO elaborated by (Bendorf, 1979) and (Recknagel, 1980). The model simulates system variables in a stratified lake with two layers epi- and hypolimnion. The conceptual model on Figure 6 shows the state variables and how they are related with processes. In each layer we have seven state variables, i.e. two inorganic nutrients (ortophosphate and nitrogen), two primary producers, one animal (zooplankton), detritus and dissolved oxygen. Mixing is applied to connect the states of a specific variable in separate layers. The process is defined in the library as *Diffusion*.

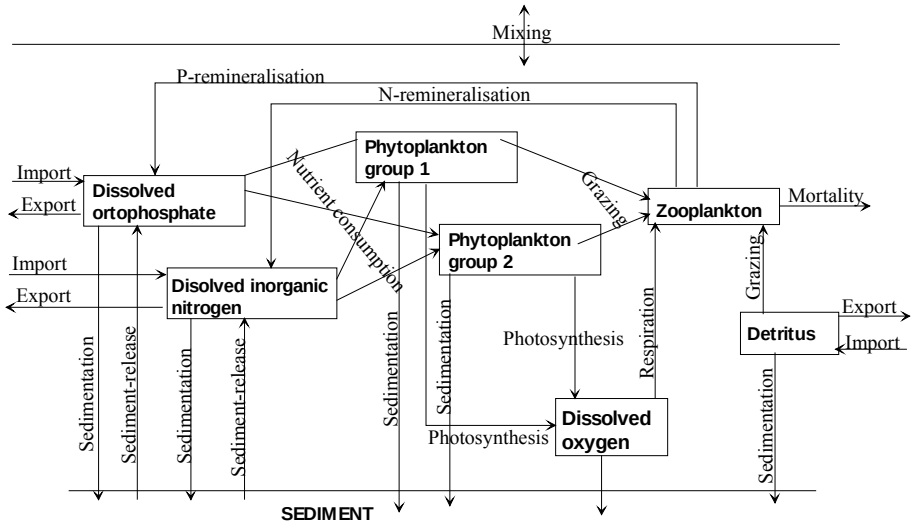


Figure 6: Conceptual model of SALMO (Benndorf, 1979 and Recknagel 1980)

The task specification for this model (two layers) is given in (Table 13). First, the independent variables are declared, i.e., external loads with inorganic nutrients (soluble inorganic phosphorus and nitrogen in the epi- and hypolimnium inflows psepi_in, pshypo_in, nsepi_in and nshypo_in), flow rates of the inflow and outflow (q_in and q_out), water temperature in epi- and hypolimnion (temp_e and temp_h) and light intensity in both layers (light_e and light_h). Next the system variables described above (seven variables in each layer) are defined. The rest of the task specification contains knowledge about the processes taking place in the system. The processes correspond to the conceptual model on Figure 6.

Table 13: Task specification for SALMO- two layers

variable Inorganic psepi_in
variable Inorganic nsepi_in
variable Inorganic pshypo_in
variable Inorganic nshypo_in
variable Oxygen o_in
variable Flow q_in
variable Flow q_out
variable Temperature temp_e
variable Temperature temp_h
variable Light light_e
variable Light light_h
system variable Inorganic ps_e
system variable Inorganic ps_h
system variable Inorganic ns_e
system variable Inorganic ns_h
system variable Primary_producer phyto1_e
system variable Primary_producer phyto2_e
system variable Primary_producer phyto1_h
system variable Primary_producer phyto2_h
system variable Animal zoo_e
system variable Animal zoo_h
system variable Oxygen o_e
system variable Oxygen o_h
system variable Detritus det_e
system variable Detritus det_h
process Inflow(ps_e, psepi_in, q_in) inflow1
process Inflow(ns_e, nsepi_in, q_in) inflow2
process Inflow(ps_h, pshypo_in, q2_in) inflow4
process Inflow(ns_h, nshypo_in, q2_in) inflow5
process Inflow(o1, o_in, q1_in) inflow3
process Outflow(ps_e,q_out) outflow1
process Outflow(ns_e,q_out) outflow2
process Outflow(o_e,q_out) outflow4
process PP_growth(phyto1_e, {ps_e,ns_e}, {temp_e},{light_e}) gr1
process PP_growth(phyto2_e, {ps_e,ns_e}, {temp_e},{light_e}) gr2
process PP_growth(phyto1_h, {ps_h,ns_h}, {temp_h},{light_h}) gr3
process PP_growth(phyto2_h, {ps_h,ns_h}, {temp_h},{light_h}) gr4
process Respiration_PP(phyto1_e, {ps_e,ns_e},{temp_e},{light_e}) resp1
process Respiration_PP(phyto2_e, {ps_e,ns_e},{temp_e},{light_e}) resp2
process Respiration_PP(phyto1_h, {ps_h,ns_h},{temp_h},{light_h}) resp3
process Respiration_PP(phyto2_h, {ps_h,ns_h},{temp_h},{light_h}) resp4
process Sediment_release(ps_e,{temp}) sed_rel1
process Sediment_release(ns_e,{temp}) sed_rel2
process Sediment_release(ps_h,{temp}) sed_rel3
process Sediment_release(ns_h,{temp}) sed_rel4
process Transformation_minus(ns_h) trans1
process Transformation_minus(ps_e) trans1
process Diffusion(ps_e,ps_h) diffusion1
process Diffusion(ns_e,ns_h) diffusion3
process Sedimentation(phyto1_e) sedimentation1
process Sedimentation(phyto2_e) sedimentation2
process Sedimentation(phyto1_h) sedimentation3
process Sedimentation(phyto2_h) sedimentation4
process Feeds_on(zoo_e, {phyto1_e,phyto2_e}, {temp_e}) feeds_on1
process Feeds_on(zoo_h, {phyto1_h,phyto2_h}, {temp_h}) feeds_on2
process Diffusion(phyto1_e,phyto1_h) diffusion5
process Diffusion(phyto2_e,phyto2_h) diffusion8
process Diffusion(det_e,det_h) diffusion9
process Excretion_A(zoo_e,{temp_e}) excretion1
process Excretion_A(zoo_h,{temp_h}) excretion2
process Excretion_A_nut(ps_e,zoo_e,{temp_e}) excretion_nut1
process Excretion_A_nut(ns_e,zoo_e,{temp_e}) excretion_nut2
process Excretion_A_nut(ps_h,zoo_h,{temp_h}) excretion_nut3
process Excretion_A_nut(ns_h,zoo_h,{temp_h}) excretion_nut4
process Mortality_A(zoo_e,{temp_e}) mortality1
process Mortality_A(zoo_h,{temp_e}) mortality2
process Mortality_A_nut(ps_e,zoo_e,{temp_e}) mortality_nut1
process Mortality_A_nut(ns_e,zoo_e,{temp_e}) mortality_nut2
process Mortality_A_nut(ps_h,zoo_h,{temp_e}) mortality_nut3
process Mortality_A_nut(ns_h,zoo_h,{temp_e}) mortality_nut4
process Entrainment(zoo_e,zoo_h) migration1
process Diffusion(zoo_e,zoo_h) diffusion9
process Entrainment_PP(phyto_e,phyto_h) entrainment1
process Entrainment_PP_ox(o1,phyto_e,phyto_h) entrainment_ox1
process Reaeration(o_e,{temp_e}) reaeration1
process Diffusion(o_e,o_h) diffusion1
process Ox_prod(o_e,{phyto1_e,phyto2_e},{ps_e,ns_e},{temp_e},{light_e}) prod1
process Ox_prod(o_h,{phyto1_h,phyto2_h},{ps_e,ns_e},{temp_e},{light_e}) pro2
process Load_sed(o_h,{temp_h}) ox_consumption

Nutrient consumption is defined through the process *PP_Growth*, while Grazing correspond to the process *Feeds_on*. P- and N- remineralisation are defined in process *Excretion_A*. The process *Mortality_A* represents the loss of zooplankton due to predation by fish, though fish is not included in the model as state variable. In

the library the PP_Growth process represent the gross growth rate of a primary producer therefore we additionally introduced the process *Respiration_PP* to account for net growth of PP. State variables in epi and hypolimnion are connected through the process *Diffusion*. The process *Transformation_minus* stands for loss of inorganic nutrient. In this case it is used for (1) loss of soluble phosphorus in epilimnion due to precipitation with calcite or other materials and (2) loss of nitrate in hypolimnion due to denitrification. Production of oxygen in photosynthesis is defined with the process *Ox_prod* and finally oxygen consumptions are put in the process *Load_sed*. This task specification is transformed into a grammar of many different model structures. However, the SALMO model can also be found among those structures.

6. Discussion

In this paper we present an approach to AM of food webs in lakes as a solid unifying framework for both handcrafting ecological models as well as their automated induction from measured data. This is enabled by encoding the existing modelling knowledge into a library of generic variables, constants and processes. Given a specification of an observed system, the AM tool transforms the knowledge in the library into specific model structures for the observed system. The structures are later optimized (according to given measurements of the system variables).

The generality of the knowledge in the library was demonstrated by generating grammars (from task specifications) that include some known lake models. Here we should point that giving those task specifications to Lagramge, produces grammars that may generate alternative models in addition to the specified ones. As we already mentioned the library contain more than one formulation for a specific process, which is reflected in the grammars. However, if the user prefers a certain model structure and do not wish other structures to be calibrated against the given data set, they can simply turn off the other formulations in the library.

The task for the simple Vollenweider's model (Table 9) generates a grammar that includes a single possible model since the model processes (inflow, outflow and sedimentation) have only one possible formulation in the library (see Table 10). This is not the case with the other two models. Imboden's model task specification contains processes with multiple formulations in the library. For example the process *PP_growth* has 20 different formulations (if light and temperature influences are included), as evident from Figure 4. In this case, since temperature and light influences are not included we have five alternatives for this process. The task given in Table 11 induces a grammar that contains all variations of the Imboden's model: from the simplest one which formulates all of the processes with a first order kinetics to the more complex that includes the Monod kinetics as evident from the segment of the grammar shown in Table 12. Further procedure of system identification with Lagramge would require a suitable data set with the measurements of all state variables in the model. Lagramge would evaluate (calibrate) all variations in order to return the variation of the Imbodens model that fits the measured data best.

SALMO (Bendorf, 1979) and (Recknagel, 1980) is the most complex model in this series. Consequently the biggest task specification (Table 13) is required to describe the model. The corresponding grammar is much bigger than the previous two.

Besides the formulations used in SALMO the grammar may also contain all others from the library, unless they are 'switched off'. In that case the grammar will only contain the SALMO model and the calibration on given measurements will only include this model. Note that SALMO in its original formulation accounts for system seasonality by adequately changing its structure. This cannot be considered in our modelling procedure in a straightforward manner. Data pre-processing is needed, in order to construct models of different structure for each season. Models for each season need to be induced on adequately prepared data for that season e.g. winter stagnation (full circulation) models need to be induced on data that represent the winter stagnation period.

In this research we illustrated the generality of the knowledge included in the library by writing task specifications that would generate grammars for some well-known models. Indeed, the knowledge is quite comprehensive, since we can generate fairly complex models, like SALMO (Recknagel, 1980). However the library has limitations, but is also a subject of extension. Major limitations of the library are connected with the present stage of Lagramge software development, i.e. (1) Regarding the physical segmentation only box models can be built, i.e. partial differential equations can not be induced and (2) Only fixed internal nutrient levels in primary producers and animals are supported. The models considered here emerged extensively in the 70-ies and early 80-ies. More recent models include a spatial dimension, since the progress in computing power facilities. Our estimation is that the library covers most of the models developed in the 70-ies and 80-ies.

6.1 Further work

In order to spread this approach of automated modelling among experts further work is aimed to developing a graphical user interface for model construction. All levels of model building supported by Lagramge will be included, namely:

- The user lets Lagramge to build the model. For this task the user must give following data to the software: physical segmentation and data of the system, a choice and type of state variables and forcing functions. Based on this information Lagramge can build a model according to the combining schemes (mass balances) coded in Lagramges library.
- User builds a model according to his or her knowledge about the system, or building models from scratch. This option requires for user to define the model structure: required data are same as above plus connections of the state variables with processes and definition of the processes. The processes and their formulations are chosen from the library. The range of parameters values can be set or the default values from the library can be used.
- The third option will offer some ready made models, like those explained in this paper. Those can be used as they are (only calibration) or they can be modified. Modification includes either a choice of another formulation for some processes, or enlargement of the search space of models by choosing more than one formulation for a specific process.

Extension of Lagramge to other related domains, like modelling of wastewater treatment plants, river modelling, which also means solving of partial differential equations is another task to be done in near future.

Finally a task that we are working on at the moment is application of the method to

real world problems. This has partly been done by Todorovski (2003), who tested Lagramge using the previous version of the knowledge library on several domains like lake Glumsoe and lagoon of Venice. Models were simple (one equation). The extension of the knowledge library enables building more comprehensive and complex models, which is by now strongly limited by the computational demands of the non-linear optimisation method.

7. Conclusions

In this research we introduced an approach to automated modelling relying on compositional induction that joins conceptual modelling and model induction from data. In particular we focused on building a comprehensive domain library of lakes ecosystem generic (elementary) models to support such modelling. The library includes many important processes used in a lake food web modelling and covers a great part of ecological models. The generality of the library was shown by reconstruction of well-known ecological models starting with the simple Vollenweider's model of one equation. Further we reconstructed two more complex models, i.e., (Imboden, 1974) and SALMO (Bendorf, 1979; Recknagel, 1980) model. Both models support physical segmentation of the lake. Moreover, SALMO includes many additional processes including sediment interaction. These examples illustrated the generality of the knowledge encoded in the library. However, the real power of the presented approach is in the searching technique among the space of all possible models, i.e. discovering the best model structure for given knowledge and data. Further work is focused on popularisation of the method among ecology experts and implementation to real world data, i.e. real system identification using the combination of domain knowledge and measured data.

References:

Baca, R., Wadel, W., Cole, C., Brandstetter, A. and Clearlock, D. (1973): EXPLORE-I: A River Basin Water Quality Model, Battelle, Inc. Pacific Northwest Laboratories, Richland, Washington.

Beck, M. (1983): A Procedure for Modeling, pp. 42. In Orlob, G. (Ed.): *Mathematical Modeling of Water Quality: Streams, Lakes and Reservoirs*, John Wiley & Sons, Chichester 0 471 10031 5.

Bendorf, J. (1979): A contribution to the phosphorus loading concept. *Int. Revue ges. Hydrobiol.* **64**, 2, 177-188.

Benz, J., Hoch, R. and Legovic, T. (2001): ECOBAS -- modelling and documentation. *Ecological Modelling* **138**, 1-3, 3-15.

Benz, J. and Hoch, R. (1997): Ein Modelldokumentationssystem. ASIM Simulationstechnik, 11. Symposium in Dortmund. , pp. 232.

Benz, J. and Knorrenschild, M. (1997): Call for a common model documentation etiquette. *Ecological Modelling* **97**, 1-2, 141-143.

Benz, J. and Voigt, K. (1996): Aufbau eines Systems zur strukturierten Suche von Informationsquellen für den Umweltschutz im Internet. Informatik für den Umweltschutz, 10. Symposium. , pp. 241.

Bertalanffy, L. (1972): The History and Starts of General System Theory. In Klir, G.J. (Ed.): *Trends in General Systems Theory*, John Wiley & Sons Inc, New York 047149190X.

Bierman, V. J., Dolan, D., Stoermer, J. G. and Smith, V. (1980): The development and Calibration of a Multi-Class Phytoplankton Model for Saginaw Bay, Lake Huron: *Great Lakes Environmental Planning Study*, Great Lakes Basin

Comission, Ann Arbor, Michigan.
Bloomfield, J., Park, R., Scavia, D. and Zahorcak, C. (1973): Aquatic Modeling in the Eastern Deciduous Forest Biome. U.S. International Biological Program, pp. 139-158: *Modeling the Eutrophication Process*, Utah State University, Logan, Utah.
Chapra, S. C. (1997): *Surface Water-Quality Modeling*. McGraw-Hill 0-07-011364-5.
Chen, C. and Orlob, C. (1975): Ecological Simulation for Aquatic Environments, pp. 588. In Patten, B. (Ed.): *Systems Analysis and Simulation in Ecology*, Academic Press Inc., New York, New York.
DeAngelis, D. L. (1992): *Dynamics of Nutrient Cycling and Food Webs*. Chapmann & Hall. London 0 412 29830 9 (HB).
Di Torro, D., O'Connor, D. and Thomann, R. (1971): A Dynamic Model of Phytoplankton Population in the Sacramento-San Joaquin Delta, pp. 131-180: *Nonequilibrium Systems in Natural Water Chemistry, Adv. Chem. Ser. 106*, American Chemical Society, Washington, D.C.
Duke, J. J. and Masch, F. (1973): Computer Program Documentation for the Stream Quality Model DOSAG3, Water Resources Engineers, Inc. For U.S. Environmental Protection Agency, Athens, Georgia, Austin Texas.
Džeroski, S. and Todorovski, L. (2003): Learning population dynamics models from data and domain knowledge. *Ecological Modelling* **170**, 2-3, 129-140.
Fasham, M. (1993): Modelling the marine biota, pp. 504. In Heimann, M. (Ed.): *The global carbon cycle*, Springer-Verlag, Berlin.
Fasham, M. (1995): Variations in the seasonal cycle of biological production in subarctic oceans. *Deep-sea Res. I* **42**, 1111-1149.
Frost, B. (1987): Grazing control of phytoplankton stock in the open subarctic Pacific Ocean: a model assesing the role of mesozooplankton, particularly the large kalanoidcopepods Neocalanus. *Mar. Ecol. Prog. Ser.* **39**, 49-68.
Hoch, R., Gabele, T. and Benz, J. (1998): Towards a standard for documentation of mathematical models in ecology. *Ecological Modelling* **113**, 1-3, 3-12.
Imboden, D. (1974): Phosphorus model of lake eutrophication. *Limnology and Oceanography* **19**, 297-304.
Imboden, D. M. and Gachter, R. (1978): A dynamic lake model for trophic state prediction. *Ecological Modelling* **4**, 2-3, 77-98.
Johanson, R., Imhoff, J. and Davis, H. (1980): Users Manual for Hydrological Simulation Program - Fortran (HSPF), Hydrocomp, Inc. For U.S. Environmental Protection Agency, Athens, Georgia, Mountain View, California.
Jørgensen, S. E. (1976): A eutrophication model for a lake. *Ecological Modelling* **2**, 2, 147-165.
Jørgensen, S., Mejer, H. and Friis, M. (1978): Examination of a Lake Model. *Ecological Modelling* **4**, 253-278.
Jørgensen, S. E. and Bendoricchio, G. (2001): *Fundamentals of Ecological Modelling*. Elsevier 0-080-44028-2.
Kompare, B. (1995): The Use of Artificial Intelligence in Ecological Modelling: *Ljubljana, FGG; Royal Danish School of Pharmacy*, FGG, Ljubljana; Royal Danish School of Pharmacy, Copenhagen, Ljubljana, Copenhagen.
Kompare, B., Todorovski, L. and Džerovski, S. (2001): Modelling and prediction of phytoplankton growth with equation discovery: case study - Lake Glumsø, Denmark. *Verh. Internat. Verein. Limnol.* **27**, 3626-3631.
Malchow, H. (1994): Nonequilibrium structures in plankton dynamics. *Ecological*

Modelling **75-76**, 123-134.
Nyholm, N. (1978): A simulation model for phytoplankton growth and nutrient cycling in eutrophic, shallow lakes. *Ecological Modelling* **4**, 2-3, 279-310.
Park, R., Collins, C., Connolly, C., Albanese, J. and MacLeod, B. (1980): Documentation of the Aquatic Ecosystem Model MS CLEANER, Rensselaer Polytechnic Institute, Center for Ecological Modeling. For US Environmental Protection Agency, Environmental Research Laboratory., Troy, New York.
Recknagel, F. (1980): Systemtechnische Prozedur zur Modellierung und Simulation von Eutrophierungs-prozessen in stehenden und gestauten Gewässern: *Sektion Wasserwesen*, TU Dresden, Dresden.
Rickel, J. and Porter, B. (1997): Automated modeling of complex systems to answer prediction questions. *Artificial Intelligence Journal* **93**, 1-2, 201-260.
Roesner, L., Giguerte, P. and Evenson, D. (1991): Computer program documentation for the Stream quality model QUAL-II, U.S. Evnironmental Protection Agency, Athens, Georgia.
Ross, A., Gurney, W. and Heath, M. (1994): A comparative study of the ecosystem dynamics of four fjords. *Limnol. Oceanogr.* **39**, 318-343.
Scavia, D. (1980): An Ecological Model of Lake Ontario. *Ecological Modelling* **8**, 49-78.
Scavia, D., Eadie, B. and Robertson, A. (1976): An Ecological Model for Lake Ontario - Model Formulation, Calibration, and Preliminary Evaluation, National Oceanic and Atmospheric Administration, Boulder, Colorado.
Scavia, D. and Park, R. A. (1976): Documentation of selected constructs and parameter values in the aquatic model cleaner. *Ecological Modelling* **2**, 1, 33-58.
Shugart, H., Goldstein, R., O'Neill, R. and and Mankin, J. (1974): TEEM: A Terrestrial Ecosystem Energy Model for Forests. *Oecol. Plant* **9**, 3, 231-264.
Smith, D. (1978): Water Quality for River-Reservoar Systems-210, Resource Management Associates, Inc. For U.S. Army Corps of Engineers, Hydrologic Engineering Center (HEC), Lafayette, California.
Smith, E. (1936): Phiotosynthesis in Relation to Light and Carbon Dioxide. *Nat. Acad. Sci. Proc.* **22**, 504-511.
Steele, J. (1965): Notes on Some Theoretical Problems in Production Ecology, pp. 393-398. In C. Goldman (Ed.): *Primary Production in Aquatic Environments.*, University of California Press, Berkeley, California.
Steele, J. and Henderson, E. (1981): A simple plankton model. *Am. Nat.* **117**, 676-691.
Talbot, P., Thébault, J.-M., Dauta, A. and De la Noüe, J. (1991): A comparative study and mathematical modelling of temperature and light on growth of three microalgae potentially useful for wastewater treatment. *Water Research* **25**, 465-472.
Thebault, J.-M. and Salencon, M.-J. (1993): Simulation model of a mesotrophic reservoir (Lac de Pareloup, France): biological model. *Ecological Modelling* **65**, 1-2, 1-30.
Todorovski, L. (2003): Using Domain Knowledge for Automated Modeling of Dynamic Systems with Equation Discovery: *Fakulteta zaraèunalništvo in informatiko*, University of Ljubljana, Ljubljana, Slovenia.
Todorovski, L., Džeroski, S. and Kompare, B. (1998): Modelling and prediction of phytoplankton growth with equation discovery. *Ecological Modelling* **113**, 1-3, 71-81.
Todorovski, L. and Džeroski, S. (1997): Declarative bias in equation discovery. 14th

International Conference on Machine Learning. , pp. 384.
Verhulst, P.-F. (1845): Recherches mathématiques sur la loi d'accroissement de la population. *Nouv. mém. de l'Academie Royale des Sci. et Belles-Lettres de Bruxelles* **18**, 1-41.
Vollenweider, R. A. (1968): The scientific basis of lake and stream eutrophication with particular reference to phosphorus and nitrogen as eutrophication factors, Organisation for Economic Cooperation and Development, Paris.
Walker, W. (1975): Description of the Charles River Basin Model: *Ch. 6 of the Final Report on the Storrow Lagoon Demonstration Plant*, Process Research, Inc. For Commonwealth of Massachusetts, Metropolitan District Commision, Cambridge, Massachusetts.

APPENDIX 1: Complete knowledge library

```
type Concentration is real
type Concentrations is set(Concentration)
type Light is real
type Lights is set(Light)
type Temperature is real
type Temperatures is set(Temperature)
type Temperatureopt is real
type Temperatureopts is set(Temperatureopt)
type Flow is real
type Area is real
type Precipitation is real
type Inorganic is Concentration
type Inorganics is set(Inorganic)
type Population is Concentration
type Populations is set(Population)
type Detritus is Population
type Detrituss is set(Detritus)
type Oxygen is Concentration
type Oxygens is set(Oxygen)
type Dom is Concentration
type Doms is set(Dom)
type Primary_producer is Population
type Primary_producers is set(Primary_producer)
type Animal is Population
type Animals is set(Animal)

function class Light_limitation(Light l) #light in J/cm2*day
function class Light_limitation_type_1() is Light_limitation
        expression l / (l + const(saturation_rate,5,30,100))
function class Light_limitation_type_2() is Light_limitation
        expression l * exp(- l / const(l_opt,150,170,300) + 1) / const(l_opt,150,170,300)
function class Light_limitation_type_3() is Light_limitation
        expression
const(foto_period,0.1,0.2,0.4)/(const(ext_coeff,0.1,0.1,0.3)*const(h,5,5,5))*ln((const(saturation_rate,
10,30,52)+l)/(const(saturation_rate,10,30,52)+l*exp(-const(ext_coeff,0.1,0.1,0.3)*const(h,5,5,5))))

function class Light_limitations(Lights ls)
        expression product({l}, l in ls, Light_limitation(l))

function class X_temp(Temperature t)
#used in Light_temp function and in type 5 of Temperature_influence
        expression (t - const(t_min,0,2,6)) / (const(t_opt,15,17,20) - const (t_min,0,2,6))

function class Temperature_influence(Temperature t)
function class Temperature_influence_type_0() is Temperature_influence
        expression t
function class Temperature_influence_type_1() is Temperature_influence
        expression (t - const(t_min,2,3,4)) / (const(t_ref,18,18,20) - const (t_min,2,3,4))
function class Temperature_influence_type_2() is Temperature_influence
        expression t / (const(t_ref,10,16.4,20))
function class Temperature_influence_type_3() is Temperature_influence
        expression pow(const(θ,1.11,1.12,1.13), (t - const(t_ref,19,19,20)))
function class Temperature_influence_type_4() is Temperature_influence
        expression exp(-2.3*( t - const(t_opt,15,16,17)) / const(t_opt,15,16,17))
function class Temperature_influence_type_5() is Temperature_influence # ASTER
        expression  const(i_max,130,140,150)*2*(1+const(b,-0.5,-0.5,-0.5))    *  X_temp(t)  /  (  1+
X_temp(t) * X_temp(t) + 2 *const(b,-0.5,-0.5,-0.5) * X_temp(t) )
function class Temperature_influence_type_6() is Temperature_influence
```

```
        expression exp(const(k,0,0.1,0.2)*t)

function class Temperature_influence_opt(Temperature t, Temperatureopt topt)
        expression exp(-const(k,0.10,0.15,0.20)*t*(t-topt))

function class Temperature_influences(Temperatures ts)
        expression product({t}, t in ts, Temperature_influence(t))

#ASTER model: one expression for light and temperature influences
function class Light_optimal(Temperature t)
        expression  const(i_max,130,140,150)*2*(1+const(b,-0.5,-0.5,-0.5))  * X_temp(t) / ( 1+
X_temp(t) * X_temp(t) + 2 *const(b,-0.5,-0.5,-0.5) * X_temp(t) )

function class X_light(Temperature t, Light l)
        expression l/Light_optimal(t)

function class Light_temp(Temperature t, Light l)
        expression  (2* (1+const(b,0,0,0)) * X_light(t,l)) / ( 1+ X_light(t,l) * X_light(t,l) + 2
*const(b,0,0,0) * X_light(t,l) )

function class Food_limitation(Inorganic c)
function class Food_limitation_type_1() is Food_limitation
        expression c / (c + const(saturation_rate, 0.0001, 0.02, 0.05))  #OP: ista polsaturacijska
konstanta pri P in N
function class Food_limitation_type_2() is Food_limitation
        expression c * c / (c * c + const(saturation_rate, 0.0005, 0.02, 0.03))
function class Food_limitation_type_3() is Food_limitation
        expression (1 - exp(-const(saturation_rate, 0.0005, 0.001, 0.03) * c))

function class Food_limitations(Inorganics cs)
function class Product() is Food_limitations
        expression product({c}, c in cs, Food_limitation(c))
function class Minimum() is Food_limitations
        expression min({c}, c in cs, Food_limitation(c))
function class Average() is Food_limitations
        expression avg({c}, c in cs, Food_limitation(c))

function class Food_pref(Population p)
        expression const(pref_fact,0,0.5,1)* p

function class Organic_food_limitations(Populations ps)
function class Organic_food_limitations_type_0() is Organic_food_limitations
        expression sum({p}, p in ps, p) / (sum({p}, p in ps, p) + const(saturation_rate, 0.001,0.1,5))
function class Organic_food_limitations_type_1() is Organic_food_limitations
        expression sum({p}, p in ps, Food_pref(p)) / (sum({p}, p in ps, Food_pref(p)) +
const(saturation_rate, 0.0001, 0.1,5))
function class Organic_food_limitations_type_2() is Organic_food_limitations
        expression sum({p}, p in ps, p)*sum({p}, p in ps, p) / (sum({p}, p in ps, p)*sum({p}, p in
ps, p) + const(saturation_rate, 0.001,0.1,5))
function class Organic_food_limitation_type_3() is Organic_food_limitations
        expression (1 - exp(-const(saturation_rate, 0.001, 0.1, 3) * sum({p}, p in ps, p)))


function class Growth_rate(Primary_producer pp, Inorganics ns, Temperatures ts, Lights ls)
function class PP_growth_unlimited() is Growth_rate
        expression const(growth_rate, 0.05, 0.4, 3)
function class PP_growth_logistic() is Growth_rate
        expression const(growth_rate, 0.05, 0.4, 3)*(1-pp/const(carr_capacity,0.01,0.1,5))
function class PP_growth_limited() is Growth_rate
        expression  const(max_growth_rate,  0.05,  0.4,  3)  *  Food_limitations(ns)  *
```

```
Temperature_influences(ts) * Light_limitations(ls)
function class Growth_rate_SALMO() is Growth_rate
        expression ((const(photx_max, 0.05, 0.4, 3)-const(photx_min, 0.05, 0.4, 0.4)) *
ts/const(topt,15,15,20)+const(photx_min, 0.05, 0.4, 0.4))*Food_limitations(ns) *
Light_limitations(ls)
function class PP_growth_topt() is Growth_rate #Bendorrichio, Topt
        expression const(max_growth_rate, 0.01, 0.1, 4) * Food_limitations(ns) *
Light_limitations(ls)*product({t1,t2}, t1 in ts, Temperature_influence_opt(t1,t2))
function class Growth_rate_type_3()is Growth_rate   #ASTER
        expression const(max_growth_rate, 0.01, 0.1, 4) * Food_limitations(ns) * product({t,l}, t in
ts and l in ls, Light_temp(t, l))

function class Filtration_rate (Populations ps, Temperatures ts)
        expression const(filtration_rate, 0.01, 0.8, 2) * Temperature_influences(ts) *
Organic_food_limitations(ps)

function class Ingestion_rate (Populations ps, Temperatures ts)
function class Ingestion1() is Ingestion_rate
        expression const(ingestion_rate, 0.01, 0.8, 1) * Temperature_influences(ts) *
Organic_food_limitations(ps)
function class Ingestion_salmo() is Ingestion_rate
        expression ((const(ing_max, 0.1, 0.8, 1)-const(ing_min, 0.05, 0.05, 0.05))*exp(-
const(r,0,0.1,2)*log(ts/const(topt,15,15,20)))+const(ing_min, 0.05, 0.05, 0.05)) *
Organic_food_limitations(ps)

function class Assimilation (Populations ps, Temperatures ts)
function class Assimilation1() is Assimilation
        expression const(assim, 0.05,0.1,1)
function class Assimilation_salmo() is Assimilation
        expression const(assim_max, 0.7, 0.8, 1)-(const(assim_max, 0.7, 0.8, 1)-const(assim_min,
0.01, 0.05, 0.2))/const(ing_max, 0.1, 0.8, 1)*Ingestion_rate(ps,ts)

function class Respiration_rate_PP (Primary_producer pp, Inorganics ns, Temperatures ts, Lights ls)
function class Respiration_rate_0() is Respiration_rate_PP
        expression const(r,0.009,0.09,0.15)
function class Respiration_rate_1() is Respiration_rate_PP
        expression const(r,0.009,0.09,0.15)*Temperature_influences(ts)
function class Respiration_rate_2() is Respiration_rate_PP
        expression pp*const(r,0.009,0.09,0.15)*Temperature_influences(ts)
function class Respiration_rate_3() is Respiration_rate_PP
        expression
const(r,0.05,0.09,0.15)*Temperature_influences(ts)+const(k,0.001,0.01,0.02)*Temperature_influenc
es(ts)*Food_limitations(ns)*Light_limitations
function class Respiration_rate_4() is Respiration_rate_PP
        expression
const(r,0.05,0.09,0.15)*Temperature_influences(ts)+const(rxmf,0.001,0.01,0.02)*Growth_rate(pp,ns,
ts,ls)
function class Respiration_salmo() is Respiration_rate_PP
        expression ((const(resp_max, 0.05, 0.4, 3)-const(resp_min, 0.05, 0.4, 0.4)) *
ts/const(topt,15,15,20)+const(photx_min, 0.05, 0.4, 0.4))
+const(rxmf,0.001,0.01,0.02)*Growth_rate(pp,ns,ts,ls)

function class Respiration_rate_A(Populations ps, Temperatures ts)
function class Respiration_A_0() is Respiration_rate_A
        expression const(r, 0.001, 0.1, 1.5)
function class Respiration_A_1() is Respiration_rate_A
        expression const(r, 0.001, 0.1, 1.5)*Temperature_influences(ts)
function class Respiration_A_2() is Respiration_rate_A
        expression const(r,0.001,0.1,1.5)* Temperature_influences(ts)+const(r,0.001,0.1,
1.5)*Temperature_influences(ts)*Organic_food_limitations(ps)
```

```
function class Respiration_A_3() is Respiration_rate_A
        expression const(r,0.001,0.1,1.5)*Temperature_influences(ts)+const(r, 0.001, 0.1,
1.5)*Ingestion_rate(ps,ts)
function class Resp_salmo() is Respiration_rate_A
        expression ((const(resp_opt, 0.07, 0.08, 0.1)-const(resp_min, 0.001, 0.005,
0.06))/const(ing_max, 0.1, 0.8, 1)*Ingestion_rate(ps,ts)+const(resp_min, 0.001, 0.005,
0.06))/const(resp_opt, 0.07, 0.08, 0.1) * ((const(resp_opt, 0.07, 0.08, 0.1)-const(resp_min, 0.001,
0.005, 0.06))*ts/const(topt,15,16,20)+const(resp_min, 0.001, 0.005, 0.06))

function class Mortality_rate_PP(Primary_producer pp, Inorganics ns,Temperatures ts, Lights ls)
function class Mortality_PP_type_0() is Mortality_rate_PP
        expression const(m, 0.003, 0.01, 1.5)
function class Mortality_PP_type_1() is Mortality_rate_PP
        expression const(m, 0.003, 0.01, 1.5)*Temperature_influences(ts)
function class Mortality_PP_type_2() is Mortality_rate_PP
        expression pp * const(m, 0.001, 0.1, 1.5)*Temperature_influences(ts)
function class Mortality_PP_type_3() is Mortality_rate_PP
        expression const(m, 0.001, 0.1, 1.5)* Temperature_influences(ts) *(1-
Food_limitations(ns)*Light_limitations(ls))
function class Mortality_PP_type_4() is Mortality_rate_PP
        expression const(m, 0.001, 0.1, 1.5)*
Temperature_influences(ts)*pp/(const(km,0.001,0.01,0.01)+pp)

function class Mortality_rate_A(Animal a, Temperatures ts)
function class Mortality_A_type_0() is Mortality_rate_A
        expression const(kd, 0.001, 0.1, 1.5)
function class Mortality_A_type_1() is Mortality_rate_A
        expression const(kd, 0.001, 0.1, 1.5) * Temperature_influences(ts)
function class Mortality_A_type_2() is Mortality_rate_A
        expression a * const(kd, 0.001, 0.1, 1.5) * Temperature_influences(ts)
function class Mortality_A_type_3() is Mortality_rate_A
        expression const(kd, 0.001, 0.1, 1.5) * Temperature_influences(ts)+ a * const(m, 0.001, 0.1,
1.5) * Temperature_influences(ts)
function class Mortality_A_hyperbolic() is Mortality_rate_A
        expression a * const(kd, 0.001, 0.1, 1.5)/(const(kd, 0.001, 0.1, 1.5)+a)
function class Mortality_A_sigmoid() is Mortality_rate_A
        expression a*a * const(kd, 0.001, 0.1, 1.5)/(pow(const(kd, 0.001, 0.1, 1.5),2)+a*a)
function class SALMO() is Mortality_rate_A
        expression const(momin, 0.001, 0.1, 0.1)+const(mot, 0.001, 0.1, 1.5)*ts*a/(const(ks, 0.001,
0.1, 0.1)+a)

function class Sod (Oxygen o, Temperatures ts) #sediment oxygen demand
function class Sod_1() is Sod
        expression const (sod,0.05, 0.1, 0.5)
function class Sod_2() is Sod
        expression const(sod, 0.1, 0.4, 0.5)*Temperature_influences(ts)*
o/(const(ks,0.022,0.5,1.22)+o)
function class Sod_3() is Sod
        expression const(sod,0.1,0.4,0.5)*o*const(h,10,10,10)+const(sod,0.1,0.4, 5)*
o/(const(ks,0.022,0.5,1.22)+o)
function class Sod_salmo() is Sod
        expression const(e, 0.4, 0.4, 0.4)*exp(const(k,0.08,0.08,0.08)*ts)*
o/(const(ks,0.022,0.022,0.022)+o)

process class Outflow(Concentration c, Flow q)
        expression c * q / const(v,7000000,7000000,7000000)

process class Inflow(Concentration c1, Concentration c2, Flow q)
        expression c2 * q / const(v,7000000,7000000,7000000)
```

```
process class Land_load(Inorganic c1, Inorganic c2, Area a)
        expression c2 * a / const(v,7000000,7000000,7000000)

process class Precip_load(Inorganic c1, Inorganic c2, Precipitation prec)
        expression prec * c2 * const(a,1470000,1470000,1470000)

process class Sedimentation (Concentration c,Temperatures ts)
process class Sedimentation_1() is Sedimentation
        expression c* const(s,0.0001,0.02, 0.3)/const(h,10,10,10)
process class Sedimentation_2() is Sedimentation
        expression c* const(s,0.0001,0.02, 0.3)/const(h,10,10,10)*Temperature_influences(ts)

process class Sedimentation_to (Concentration c1, Concentration c2)
        expression c1* const(s,0.0001,0.02, 0.3)/const(h,10,10,10)

process class Entrainment(Concentration c1, Concentration c2)
        expression c2* const(s,0.001,0.3, 0.5)*const(v,0.01,0.1,0.5)/const(h,10,10,10)

process class Diffusion(Concentration c1, Concentration c2)
        expression (c2-c1)* const(diff,0.001,0.3, 0.5)/const(h,5,5,5)

process class PP_growth(Primary_producer pp, Inorganics ns, Temperatures ts, Lights ls)
        expression pp * Growth_rate(pp,ns,ts,ls)

process class Feeds_on(Animal a, Populations ps, Temperatures ts)
process class Filtration() is Feeds_on
        expression a * Filtration_rate(ps,ts)
process class Ingestion() is Feeds_on
        expression a * Ingestion_rate(ps,ts)* 1/sum({p}, p in ps, Food_pref(p))

process class Respiration_PP (Primary_producer pp, Inorganics ns,Temperatures ts, Lights ls)
        expression pp*Respiration_rate_PP(pp,ns,ts,ls)

process class Respiration_A(Animal a, Populations ps, Temperatures ts)
        expression a * Respiration_rate_A(ps,ts)

process class Mortality_PP(Primary_producer pp, Inorganics ns,Temperatures ts, Lights ls)
        expression pp * Mortality_rate_PP(pp,ns,ts,ls)

process class Mortality_A(Animal a, Temperatures ts)
        expression a * Mortality_rate_A(a,ts)

process class Excretion_PP (Primary_producer pp, Temperatures ts) # =respiration_PP
        expression pp * const(e, 0.001, 0.1, 1.5)*Temperature_influences(ts)

process class Excretion_A (Animal a, Temperatures ts)
        expression a * const(e, 0.001, 0.1, 1.5)*Temperature_influences(ts)

process class Decomposition (Detritus d)
        expression d * const(kr, 0.001, 0.1, 1)

process class Hydrolysis (Dom d, Temperatures ts)
        expression d * const(kr, 0.001, 0.1, 1)*Temperature_influences(ts)

process class Transformation_minus (Inorganic i, Temperatures ts) # nitrification for NH4,
denitrification for NO3
        expression i * const(k, 0.001, 0.1, 1)*Temperature_influences(ts)

process class Transformation_plus (Inorganic i1,Inorganic i2, Temperatures ts) # nitrification for
NO3
```

```
        expression i2 * const(k, 0.001, 0.1, 1)*Temperature_influences(ts)

process class Sediment_release(Concentration c,Temperatures ts)
        #expression                                              const(areal_release,0.001,0.3,
0.5)/const(h,15,15,15)*Temperature_influences(ts)
        expression          (const(areal_release_min,0.001,0.3,          0.5)+const(slope,0.001,0.3,
0.5)*ts)/const(h,15,15,15)

process class Migration(Animal a1, Animal a2)
        expression a2 * const(vm, 0.01, 0.1, 1.5)/const(h,10,10,10)

process class Excretion_A_nut (Inorganic i, Animal a, Temperatures ts)
        expression const(conv, 0.001, 0.1, 1.5)*Excretion_A(a,ts)

process class Mortality_A_nut (Inorganic i, Animal a, Temperatures ts)
        expression const(conv, 0.001, 0.1, 1.5)*Mortality_A(a,ts)

process class Respiration_PP_nut (Inorganic i,Primary_producer pp,Inorganics ns,Temperatures ts,
Lights ls)
        expression const(conv, 0.001, 0.1, 1.5)*Respiration_PP(pp,ns,ts,ls)

process class Reaeration (Oxygen o, Temperatures ts)
        expression const(ox_sat,14.26 ,14.26, 14.26)*exp(const(ox_sat,0.022,0.022,0.022)*ts)

process class Ox_prod (Oxygen o, Primary_producers pps,Inorganics ns, Temperatures ts, Lights ls)
        expression const(y,0.1,1, 5)* sum({pp},pp in pps, PP_growth(pp,ns,ts,ls))

process class Sediment_cons (Oxygen o, Temperatures ts) #loading of the water body with organic
due to ox. cons. by sediment
        expression Sod(o,ts)/ const(h, 10, 10, 10)

process class Decomposition_ox (Oxygen o, Detritus d)
        expression const(y, 0.1, 0.1, 4)*Decomposition(d)

process class Sedimentation_to_ox(Oxygen o,Concentration pp1,Concentration pp2)
        expression const(conv,0.001,0.3, 0.5)*Sedimentation_to(pp1,pp2)

process class Respiration_PP_ox (Oxygen o,Inorganic i,Primary_producer pp,Inorganics
ns,Temperatures ts, Lights ls)
        expression const(conv, 0.001, 0.1, 1.5)*Respiration_PP(pp,ns,ts,ls)

process class Respiration_A_ox(Oxygen o, Animal a, Populations ps, Temperatures ts)
        expression const(conv, 0.001, 0.1, 1.5)*Respiration_A(a,ps,ts)


combining scheme Lake(Inorganic i)
        time_deriv(i) =
                + sum({c2, q}, true, Inflow(i,c2,q))
                + sum({c2,a}, true, Land_load(i,c2,a))
                + sum({c2, prec}, true, Precip_load(i,c2, prec))
                - sum({q_out}, true, Outflow(i,q_out))
                + sum({d}, true, const(conversion_factor, 0,0.1, 1)*Decomposition (d))
                + sum({i1}, true, Diffusion(i,i1))
                - sum({i1}, true, Diffusion(i1,i))
                + sum({a,ts}, true, Mortality_A_nut(i,a,ts))
                + sum({a,ts}, true, Excretion_A_nut(i,a,ts))
                + sum({pp,ts,ns,ls}, t0rue,Respiration_PP_nut(i,pp,ns,ts,ls))
                - sum({}, true, Transformation_minus(i))
                + sum({i2}, true, Transformation_plus(i,i2))
                + sum({ts}, true, Sediment_release(i,ts))
```

```
                - sum({ts}, true, Sedimentation(i,ts))
                - sum({i1}, true, Sedimentation_to(i,i1))
                + sum({i1}, true, Sedimentation_to(i1,i))
                - sum({pp, food, ts, ls}, i in food, const(conversion_factor, 0.0005,0.002, 0.009) *
PP_growth(pp, food, ts, ls))

combining scheme Lake(Oxygen o)
        time_deriv(o) =
                + sum({ts}, true, Reaeration(o,ts))
                + sum({o2, q}, true, Inflow(o,o2,q))
                + sum({pp,ns,ts,ls}, true, Ox_prod(o,pp,ns,ts,ls))
                - sum({q_out}, true, Outflow(o,q_out))
                + sum({o1}, true, Diffusion(o,o1))
                - sum({o1}, true, Diffusion(o1,o))
                - sum({pp,ts,ns,ls}, true, Respiration_PP_ox(o,pp,ns,ts,ls))
                -       sum({a,ps,ts},      true,      const(conversion_factor,      0,0.1,
5)*Respiration_A_ox(o,a,ps,ts))
                - sum({ts}, true, Sediment_cons (o,ts))
                - sum({d}, true, Decomposition_ox (o,d))
                -       sum({pp1,pp2},      true,      const(conversion_factor,      0,0.1,
5)*Sedimentation_to_ox(o,pp1,pp2))

combining scheme Lake(Primary_producer pp)
        time_deriv(pp) =
                + sum({food, ts, ls}, true, PP_growth(pp, food, ts, ls))
                - sum({ns,ts,ls}, true, Respiration_PP(pp,ns,ts,ls))
                - sum({ns,ts,ls}, true, Mortality_PP(pp, ns,ts,ls))
                - sum({}, true, Outflow(pp))
                - sum({ts}, true, Sedimentation(pp,ts))
                - sum({pp1}, true, Sedimentation_to(pp,pp1))
                + sum({pp1}, true, Sedimentation_to(pp1,pp))
                + sum({pp1}, true, Diffusion(pp,pp1))
                - sum({pp1}, true, Diffusion(pp1,pp))
                - sum({a, food, ts}, pp in food, Feeds_on(a, food, ts))*Food_pref(pp)

combining scheme Lake(Detritus d)
        time_deriv(d) =
                + sum({pp,ns,ts,ls}, true, Mortality_PP(pp,ns,ts,ls))
                + sum({a,ts}, true, Mortality_A(a,ts))
                - sum({ts}, true, Sedimentation(d,ts))
                - sum({}, true, Outflow(d))
                - sum({}, true, Decomposition (d))
                + sum({d1}, true, Diffusion(d,d1))
                - sum({d1}, true, Diffusion(d1,d))
                - sum({a, food, ts}, d in food, Feeds_on(a, food, ts))*d

combining scheme Lake(Dom do)
        time_deriv(i) =
                - sum({q_out}, true, Outflow(do,q_out))
                + sum({d}, true, const(conversion_factor, 0,0.1, 1)*Decomposition (d))
                + sum({a,ts}, true, const(conversion_factor, 0,0.1, 1)*Excretion_A(a,ts))
                - sum({do}, true, Hydrolysis (do))
                + sum({do1}, true, const(conversion_factor, 0,0.1, 1)*Entrainment(do,do1))
                + sum({do1}, true, Diffusion(do,do1))
                - sum({do1}, true, Diffusion(do1,do))


combining scheme Lake(Animal a)
        time_deriv(a) =
                + sum({p, food, ts}, p in food, Assimilation(food,ts)*Feeds_on(a, food, ts) *
```

```
Food_pref(p))
                - sum({ts}, true, Mortality_A(a,ts))
                + sum({a1}, true, Migration(a, a1))
                - sum({ts}, true, Excretion_A(a,ts))
                - sum({a1, food, ts}, a in food, Feeds_on(a1, food, ts)) * a
```

The use of the expert modelling knowledge in the procedure of automated modelling

Nataša Atanasova[1], Ljupčo Todorovski[2], Sašo Džeroski[2], Boris Kompare[1]

1 Faculty of Civil and geodetic Engineering, University of Ljubljana, Slovenia
[2] Jožef Štefan Institute, Slovenia.

Abstract
This paper deals with incorporation of the expert modelling knowledge into automated model induction from data. The knowledge is incorporated in form of knowledge library. We analyse the use of two similar, yet of different complexity, knowledge libraries from the domain of aquatic food-web modelling. The first is a simple one developed in order to illustrate the applicability of the automated modelling method. Some good models from real data domains, i.e., for lake Glumsoe and Lagoon of Venice, have been discovered with this library. Yet, the models' structures do not completely follow the expert knowledge, due to some inconsistencies in the knowledge library. The second library is more comprehensive, it was estimated to cover a great part of the existing modelling knowledge from this domain. This library was used on the same domains in order to compare the obtained resulting models. The models are analysed in their accuracy and structure. The results from Lagoon of Venice show similar accuracy of the models discovered with both libraries. But the structure of the models obtained with the complex library is more in accordance with the expert modelling knowledge. For the lake Glumsoe models of correct structure were obtained already with the simple library. Using the complex library Lagramge found slightly different model structure, which also performs better in comparison with the measured data.

1. Introduction

Conceptual mathematical modelling of ecosystems is very complex domain, where the existing modelling knowledge is still quite incomplete. Processes that happen in nature are sometimes difficult to understand and therefore difficult to be put in equations. Though a tremendous work has been done in this field (e.g. Jorgensen and Bendoricchio, 2001; De Angelis, 1992; Chapra, 1997; and so on), scientists search for alternative methods that can be of help in building models. Such is the field of automated modelling (AM), comprising several different methods, which assists scientists with this task. Data-driven methods are used for building models without the necessity to introduce any domain knowledge in the process of model construction. These models are so called black-box models, i.e. their structure can not be interpreted by domain experts. Some of the data driven methods, which mostly belong to the field of machine learning (ML) are capable of building so called semi-transparent models. This means that they can be partly explained and understood by an expert. Successful applications of different machine learning techniques in ecology can be found for

example in (Kompare, 1995; Kompare and Džeroski, 1995; Dzeroski and Todorovski, 1993). However, the fact remains that they are induced from data, without incorporating any domain knowledge in the induction procedure. Unlike these methods compositional modelling methods are aimed to help scientists in building mathematically correct models' structures. The models are built by composing model fragments, commonly encoded in a library, into an adequate model of the entire system. Main elements of the compositional modelling framework are the knowledge base, the specification of the observed system, and an algorithm capable of composing and evaluating different models.

Recently, an approach to automated modelling, based on equation discovery methods has been developed (Todorovski and Džeroski, 2001; Langley et al. 2002; Todorovski, 2003). Unlike the other AM methods this one enables introduction of the expert (domain) knowledge in the procedure of automated model induction (equation discovery). As a result, the method discovers a set of models (equations) that follow the basic principles in the domain of interest. Thus, the developed AM framework, called Lagramge 2.0, discovers both, the structure and the parameters of the model. In the early days of the development of this tool (Todorovski and Džeroski, 1997) the knowledge had to be provided as an explicit specification of the space of candidate models. Now, the tool allows the user to provide higher-level (generic) domain knowledge about building mathematical models of complex real-world systems (Todorovski, 2003).

In order to be used in the induction procedure the modelling knowledge needs to be properly coded in a knowledge library. Lagramge supports modelling with ordinary differential equations (ODEs), which are one of the most commonly used tool for building ecological models. This kind of modelling requires background knowledge about the ecological processes that take place in ecosystems.

Todorovski (2003) introduced a knowledge library for building simple models in the domain of population dynamics. The main purpose of this library was to confirm the applicability of the proposed induction method. With this library he discovered some simple models from real-world aquatic ecosystems, i.e. for the lake Glumsoe and Lagoon of Venice. However, the library is rather simplistic, and it can not be used for inducing more complex models, i.e. it does not cover properly all aspects of knowledge from the domain of lake food web modelling. Atanasova et al. (2005) developed a comprehensive knowledge library that was estimated to cover great part of the existing modelling knowledge from this domain. Further, the models discovered from this library are structurally correct according to the expert modelling knowledge. Known ecological models of different complexity can be derived from the library, such as the simple Vollenweider's model (Vollenweider, 1968) or the fairly complex SALMO (Bendorf, 1979 and Recknagel, 1980). For details see Atanasova et al. (2005).

The knowledge gathered in the library has great influence on the results, i.e. the models revealed by Lagramge, especially on their structure. This is the most important issue of the developed method (Todorovski, 2003), since the revealed models should be consistent with the domain knowledge. Note that many black box models can be accurate, but they can not be explained by experts. After all it is known that models with consistent structure may not be the most accurate comparing with the data on which they are induced, but their performance on new data is usually much better then the performance of the black-box models.

This paper gives an explanation on how the two developed background knowledge libraries influence the resulting models, obtained by the automated modelling procedure. Thus, we compare (1) the simple knowledge library from the domain of aquatic population dynamics (Todorovski, 2003) and (2) fairly complex library from the same domain (Atanasova, 2005). In this paper we will point the limits as well as the inconsistencies between the knowledge in the library and the expert lake-modelling knowledge.

The paper is organized as follows: in the next chapter we briefly explain the automated modelling framework. In section three we introduce the formalism of coding the modelling knowledge in the library and how it is further used in model induction. Section four gives a description of the two knowledge libraries, i.e. the simple and the complex. Finally in section five we evaluate the libraries on real data, followed by some conclusions.

2. The method: automated modelling framework

The procedure of automated modelling using the submitted, i.e. measured and suitably (re)interpreted data (see Kompare, 1995) on the one side and the background knowledge on the other side is shown in Figure 1. The modelling knowledge is gathered in a library of domain-specific knowledge. Next, modelling task has to be defined. This is done (in present version still manually) by user's specification of the observed system variables and processes that are expected to influence the behaviour of the system. Given a specification of modelling task at hand, Lagramge preprocessor can transform the high-level knowledge from the library into an operational form of a grammar. This grammar now completely specifies the space of candidate models of the observed system. This is illustrated in the left-hand side of Figure 1.

Once we have the grammar, we can use equation discovery system Lagramge to heuristically search through the space of candidate models, match each of them to submitted data by fitting the values of the constant parameters. These models evaluated (sorted) by two error measurements, i.e. mean square error (MSE) and MDL are the output of Lagramge. Further details about the modeling framework from Figure 1 can be found in (Todorovski, 2003).

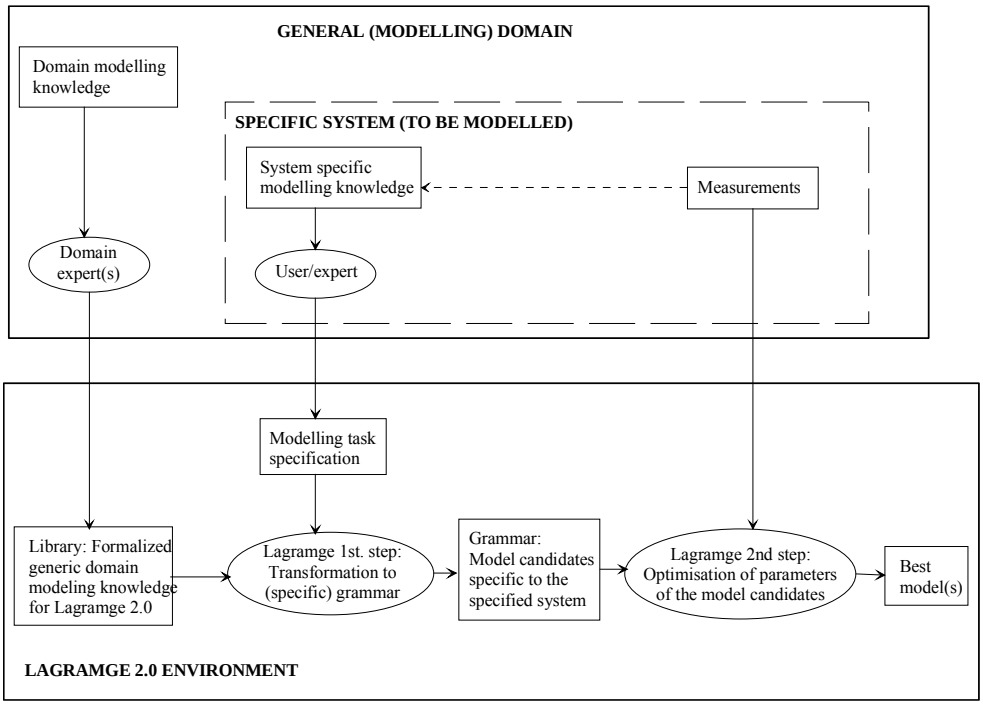


Figure 1: An automated modeling framework based on the integration of domain-specific modeling knowledge in the process of equation discovery

3 Domain specific modelling knowledge - The formalism

In order to be used in the procedure of automated modelling (Figure 1) the knowledge needs to be appropriately coded. Todorovski (2003) developed formalism for encoding the domain knowledge. The formalism supports modelling with ordinary differential equations (ODE) by following the mass conservation principle (e.g. Jorgensen and Bendoricchio, 2001; De Angelis, 1992; Chapra, 1997; and so on). Modelling with ODEs is illustrated on a simple example in the next section. Using the developed formalism he created a simple knowledge library for modelling population dynamics.

3.1 Domain knowledge about primary producer dynamics

We will illustrate the formalism for encoding the domain knowledge on a simple example from the domain of lake modelling. Suppose we want to encode the knowledge about modelling of primary producer (PP) dynamics. In general, the PP dynamics, i.e. temporal change of primary producer concentration is stated as follows (1):

$$\frac{dPP}{dt} = (\text{growth}) - (\text{non\_predatory\_losses}) - (\text{predatory\_losses}) \tag{1}$$

The equation (1) represents a mass balance of the *PP* concentration, which is a suitable combination of biochemical processes. The mass of *PP* increases due to the process of

growth and decreases due to non-predatory and predatory losses. Non-predatory losses are for example respiration, sedimentation and natural mortality. A predatory loss represents for example grazing of zooplankton on *PP*. To keep the example simple and clear to the reader we will suppose only the respiration process as a primary producer loss. The mass balance is now rewritten as (2):

$$\frac{dPP}{dt} = \text{growth-respiration} \tag{2}$$

Inorganic nutrients represent a food for primary producers. Therefore the growth of *PP*, or the increase of the mass of *PP* affects the concentration (it decreases) of the nutrients in the system. In contrast, by respiration *PP* releases inorganic nutrients. Thus, this process causes an increase in nutrients concentration. Equation represents temporal change of nutrient concentration or the mass balance of inorganic nutrient. Note that in reality the mass balance contains many more processes, such as inflow or outflow of nutrients or PP, along with their storage within the observed system. The mass balance here is kept simple in order to make the illustration of encoding the modelling knowledge as clear as possible. The constants in the equation (3) represent the conversion factors of biomass to nutrient, i.e. the stoichiometric ratio between the biomass *PP* and a specific inorganic nutrient *Nut*.

$$\frac{dNut}{dt} = -\text{const} \cdot \text{growth-const} \cdot \text{respiration} \tag{3}$$

Conceptually, we can present our modelling knowledge as in Figure 2. The boxes represent the system states or state variables (primary producer and inorganic nutrient), while the arrows represent the processes that influence the states. The process *growth* is pointed to *PP* which means that this process have positive influence on *PP*, i.e. it contributes to the *PP* mass (or concentration), while *respiration* has negative influence on *PP* and positive on *Nut*. Thus, our modelling knowledge contains two processes (growth and respiration) and two differential equations that combine those processes into a model for primary producer and inorganic nutrient.

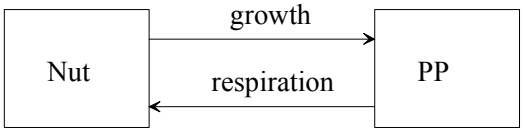


Figure 2: Conceptual model of a primary producer (PP) and a nutrient (Nut) dynamics

Note that the scheme in Figure 2 represents a generalized knowledge. The variables in the boxes represent **types** of variables. This knowledge can be used for modelling more

specific systems, for example a system where we observe two species of primary producers and three inorganic nutrients (five state variables).

Formulations of the processes - from conceptual to mathematical model

In order to quantify the conceptual knowledge explained above we need to formulate the biochemical processes with mathematical expressions. The primary producer growth process can be formulated by the exponential model (4), where growth process is only dependant on the primary producer concentration.

$$growth = \mu \cdot PP \tag{4}$$

where PP is a primary producer concentration [mass/volume] and μ is primary producer growth rate [1/time].

In reality the growth is influenced also by nutrients, temperature and light. In this example we will show the formulation where the growth of primary producers is limited (influenced) by the nutrients (equation 5):

$$\mu = \mu_{\max} \cdot f(P) \tag{5}$$

where f(P) is nutrient limitation function of growth (P is the limiting nutrient).

Most ecological models formulate the nutrient limitation on the growth rate with the Monod's expression:

$$f(P) = \frac{P}{k + P} \tag{6}$$

If the growth is limited by more then one nutrient then the total influence (limitation) can be expressed by the product of the limiting functions:

$$f(C,N,P) = f(C) \cdot f(N) \cdot f(P) = \frac{C}{k + C} \cdot \frac{N}{k + N} \cdot \frac{P}{k + P} \tag{7}$$

The respiration process can be formulated with first order kinetics. Loss of primary producer due to respiration is equal to:

$$respiration = -k \cdot PP \tag{8}$$

where k is the rate coefficient [1/time].

To summarise, our knowledge about modelling of a primary producer dynamics

comprises two generalised differential equations, where two processes are combined – growth and respiration. The growth process can have three different formulations – exponential, logistic (limited by population) and limited (by nutrients). Exponential and logistic growths are functions of the primary producer concentration only, whereas the limited growth is function of both, the primary producer concentration and the inorganic nutrients concentrations. Respiration has just one formulation and is dependant on the *PP* concentration.

3.2 Encoding the knowledge on primary producer and nutrient dynamics

Domain specific modelling knowledge in the library is formalized in terms of (1) taxonomy of variable types, (2) taxonomy of basic processes that govern the behaviour of aquatic ecosystems, (3) alternative models of the basic processes, and (4) knowledge how to combine models of individual processes into a model of the entire ecosystem. To encode the simple knowledge presented in previous Section we first need to declare the variable types in the system (Table 1). We have two variables – inorganic nutrient and primary producer, both expressed as concentrations [mass/volume]. Therefore we can declare one generic variable type *Concentration* and two subtypes, i.e., *Inorganic*, representing the inorganic nutrients for primary producers and *Primary producer*, representing the primary producers. If we want to model interaction between many species (for example primary producer grazing on more then one nutrient) we need to declare sets of variables (lines 3 and 5).

Table 1: Taxonomy of variable types for the system presented in Figure 2

```
1:  type Concentration is real
2:  type Primary_producer is Concentration
3:  type Primary_producers is set(Primary producer)
4:  type Inorganic is Concentration
5:  type Inorganics is set(Inorganic)
```

Next step is taxonomy of process classes. Each process class represents a class of basic process formulations (models). In our case we have two process classes, i.e. *Growth_PP* and *Respiration_PP*. The first has two sub-classes (exponential and limited) and the second has one sub-class (Table 2).

The definition of each process class consists of: types of the variables involved and declaration of the process models (Todorovski, 2003). The first part specifies the types of variables that can influence or be influenced by the processes in the class (first line in the process definition). The types of variables in the process Growth_PP are primary producer (*pp*) and Inorganics (*cs*), which represents a set of food sources on which the primary producer pp depends. If a variable in a process has sub-types, then all subtypes of that variable will be influenced by that specific process. For example, if a variable of type concentration is involved in a process class, then sub-types variables of the type concentration will be or will have influence on that process.

Table 2: Taxonomy of process classes for the system presented in Figure 2

```
1:  process class Growth_PP (Primary_producer pp, Inorganics cs)
2:          process class Exponential() is Growth_PP
3:                  expression const(growth_rate,0,0.5,2)*pp
4:          process class Limited() is Growth_PP
5:                  expression const(growth_rate,0,0.5,2)*pp*product({c}, c in cs,
6:          c/(const(saturation,0,1,2)+c)

7:  process class Respiration_PP (Primary_producer pp)
8:          process class Exponential() is Respiration_PP
9:                  expression const(resp_rate,0,0.5,2)*pp
```

Declaration of process models specifies the equation template in the process class. The process class has as many subclasses as there are models for the class. The equation template can include variables involved in the process and generic constant parameters, which are specified with the symbol *const(name, lower_bound, initial_value, upper_bound)*. The generic parameter constants can be later fitted to the measurements. Note the product term in the third sub-class of the Growth_PP process. The term is used to multiplicatively combine the primary producer food limitation terms as presented in equation (7).

The library language formalism also supports declaration of function classes. These are beneficial when we want to declare an influence in the process class, which can have more then one formulation. For example, the third subclass in the Growth_PP process class contains a nutrient limitation expression which is formulated using the Monod's expression. From the background knowledge we know for at least two more expressions that can be used for nutrient limitation functions. Thus, we can define a function class *Food_limitation* containing all expressions that can be used in the formulation of the process Growth_PP (Table 3).

Table 3: Definition of the function class Food_limitation

```
1:  function class Food_limitation(Inorganic c)
2:          function class Food_limitation_type_1() is Food_limitation
3:                  expression c / (c + const(saturation_rate, 0, 0.02, 10))
4:          function class Food_limitation_type_2() is Food_limitation
5:                  expression c * c / (c * c + const(saturation_rate, 0, 0.02,10))
```

In order to incorporate the Food_limitation function class into Growth_PP process class we must rewrite the Growth_PP process class as shown in Table 4.

Table 4: Final definition of the process class Growth_PP

```
process class Growth_PP (Primary_producer pp, Inorganics cs)
        process class exponential is Growth_PP
                expression const(growth_rate,0,0.5,2)*pp
        process class limited is Growth_PP
                expression const(growth_rate,0,0.5,2)*pp*product({c},c in cs,
        Food_limitation(c))
```

Finally combining schemes are used to combine the process classes into a model of the whole system. Each specific combining scheme represents a mass balance for a specific type of state variable. In our example we used two types of state variables (Inorganic and Primary_producer). Thus we need two combining schemes (Table 5).

Table 5: Combining schemes for combining the process classes into a model of the whole system

```
combining scheme Lake(Inorganic i)
        time_deriv(pp) =
                - sum({pp}, true, const(conv_fac,0,0.01,1)*Growth_PP(pp, i))
                + sum({pp}, true, const(conv_fact,0,0.01,1)*Respiration_PP(pp))

combining scheme Lake(Primary_producer pp)
        time_deriv(pp) =
                + sum({food}, true, Growth_PP(pp, food))
                - sum({}, true, Respiration_PP(pp))
```

Note the use of the aggregation function *sum*. The function in the first combining scheme is used to summarize all expressions of the Growth_PP process, in which an arbitrary primary producer pp consumes the nutrient i. The use of the aggregation function seems unnecessary in the second combining scheme, since the process Growth_PP here represents only the growth of the primary producer pp. In such cases the use of the aggregation functions is beneficial when the process is not present in the observed system. In that case the value of the term will be 0, thus no influence on the mass balance.

3.3 The use of the knowledge library in automated modelling

3.3.1 Task specification

In the task specification the expert (user) introduces the knowledge for a particulate observed ecosystem to the model discovery tool, which is further used in the equation discovery procedure. An example that can be modelled with the knowledge encoded above is shown in Figure 3. Suppose we want to model phytoplankton and nutrients dynamics in a lake. To our knowledge the phytoplankton concentration is increasing due to consumption of two nutrients (p and n) and decreasing due to respiration. In contrast the nutrients concentrations are decreasing due to phytoplankton growth and

increasing due to respiration.

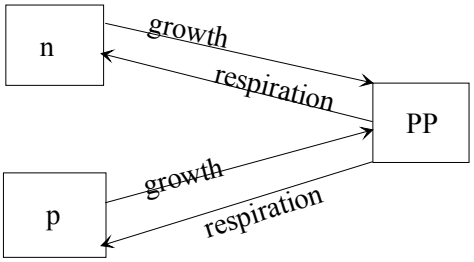


Figure 3: Conceptual model for primary producer feeding on two nutrients

The task specification includes declaration of the observed variables and processes in the system to be modelled. The variables in the system are declared, by giving the variable type and the variable name:

variable variable_type 'variable_name'

A process in the system is defined by the word *process*, followed by the process' name and the process' arguments:

process 'process_name' (arguments) **notation**

The task specification for this system is given in Table 6. It contains declarations of three system variables (n, p and phyto) and two processes Growth_PP, representing the phytoplankton growth and Respiration, for phytoplankton respiration.

Table 6: Modelling task specification for the system in Figure 3

```
1:  variable Inorganic n
2:  variable Inorganic p
3:  variable Primary producer phyto

4:  process Growth_PP (phyto,{n,p}) gr
5:  process Respiration (phyto) resp
```

3.3.1 Transforming the task into candidate models of the system

The task is further transformed into a grammar of all possible models for this system. First, the combining schemes are applied to all system variables. Following equations for the temporal change of the variable types are obtained:

$$n' = -\text{const}(conv\_fact,0,0.01,1)\cdot \text{Growth}(phyto,\{n,p\}) + \text{const}(conv\_fact,0,0.01,1)\cdot \text{Respiration}(phyto)$$
$$p' = -\text{const}(conv\_fact,0,0.01,1)\cdot \text{Growth}(phyto,\{n,p\}) + \text{const}(conv\_fact,0,0.01,1)\cdot \text{Respiration}(phyto)$$
$$phyto' = +\text{Growth}(phyto,\{n,p\}) - \text{Respiration}(phyto)$$

Further, the process classes definitions are transformed into suitable model structures, called the grammar. Transformation of the two process classes is given in table (Table 7). The process class Growth(phyto,{n,p}) is transformed into two structures (lines 1 and 2). The first has one possible formulation, whereas the second has four, due to the four formulations of the function Food_limitation(n,p) (see lines 3 to 11). Thus the process class Growth(phyto,{n,p}) has five possible formulations. Because the respiration process class is transformed into a single model, this task specification is transformed into five possible model structures.

Table 7: Transformation of the process class Growth from the task in Table 6 into model structures

1:	1. Growth(phyto,{n,p})= const*phyto
2:	2. Growth(phyto,{n,p}) = const*phyto*Food_limitation(n,p)
3:	3. Food_limitation(n,p) = Food_limitation(n)* Food_limitation(p)
4:	4. Food_limitation(n) = Food_limitation_type_1(n)
5:	5. Food_limitation(n) = Food_limitation_type_2(n)
6:	6. Food_limitation_type_1(n) = n/(const + n)
7:	7. Food_limitation_type_2(n) = n*n/(const + n*n)
8:	8. Food_limitation(p) = Food_limitation_type_1(p)
9:	9. Food_limitation(p) = Food_limitation_type_2(p)
10:	10. Food_limitation_type_1(p) = p/(const + p)
11:	11. Food_limitation_type_2(p) = p*p/(const +p*p)
12:	12. Respiration (phyto)= const(resp_rate,0,0.5,2)*phyto

In Table 8 we give the grammar, i.e. the four candidate model structures, obtained with this task specification. Coefficients k1 to k8 represent the parameters of the model.

Given the observed values of the system variables over time, the automated modelling method chooses the model that fits the measurements best (see Figure 1). This best fit model selection is done by very intensive work of simultaneous parameter optimization for each of the models, see Table 8

Table 8: Candidate model structures for the task specification in Table 6

Model 1 $n' = -k1 \cdot k2 \cdot phyto + k3 \cdot k4 \cdot phyto$ $p' = -k5 \cdot k2 \cdot phyto + k6 \cdot k4 \cdot phyto$ $phyto' = k2 \cdot phyto - k4 \cdot phyto$	Model 2 $n' = -k1 \cdot k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p}{k8+p} + k3 \cdot k4 \cdot phyto$ $p' = -k5 \cdot k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p}{k8+p} + k6 \cdot k4 \cdot phyto$ $phyto' = k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p}{k8+p} - k4 \cdot phyto$
Model 3 $n' = -k1 \cdot k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p}{k8+p} + k3 \cdot k4 \cdot phyto$ $p' = -k5 \cdot k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p}{k8+p} + k6 \cdot k4 \cdot phyto$ $phyto' = k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p}{k8+p} - k4 \cdot phyto$	Model 4 $n' = -k1 \cdot k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p^2}{k8+p^2} + k3 \cdot k4 \cdot phyto$ $p' = -k5 \cdot k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p^2}{k8+p^2} + k6 \cdot k4 \cdot phyto$ $phyto' = k2 \cdot phyto \cdot \frac{n}{k7+n} \cdot \frac{p^2}{k8+p^2} - k4 \cdot phyto$
Model 5 $n' = -k1 \cdot k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p^2}{k8+p^2} + k3 \cdot k4 \cdot phyto$ $p' = -k5 \cdot k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p^2}{k8+p^2} + k6 \cdot k4 \cdot phyto$ $phyto' = k2 \cdot phyto \cdot \frac{n^2}{k7+n^2} \cdot \frac{p^2}{k8+p^2} - k4 \cdot phyto$	

4 Knowledge libraries about lake modelling

Using the formalism described in the previous chapter Todorovski (2003) developed a simple library for modelling population dynamics. The main purpose of the library was to illustrate and prove the applicability of the automated modelling method (Figure 1), thus it enables building simple models from this domain. Due to the simplicity, some inconsistencies in the model structures derived from the library can be found. In order to point on these issues we will analyse three generic process classes in the library - Growth, Decay and Feeds_on. The taxonomy of these process classes is given in Table 9. The process class Growth describes growth of a single population. It has two sub-classes, i.e. exponential and logistic growth. Feeds_on describes predator prey interactions. It accounts for limited predation capacity by using four different limitation (saturation) functions (Table 10).

Table 9: Taxonomy of selected process classes in the population dynamics knowledge library (Todorovski, 2003)

```
process class Growth(Population p)
        process class Exponential_growth() is Growth
                expression const(growth_rate,0,0.1,10) * p
        process class Logistic_growth() is Growth
                expression const(gr_rate,0,0.1,10) * p * (1 - p /
        const(capacity,0,0.1,10))

process class Decay(Population p)
        process class Exponential_decay() is Decay
                expression const(decay_rate,0,0.1,10) * p

process class Feeds_on(Population p, Concentrations cs)
        condition p not in cs
                expression p * product({c}, c in cs, Saturation(c))
```

Table 10: Definition of the function class Saturation (Todorovski, 2003)

```
function class Saturation(Concentration c)
        function class No_saturation() is Saturation
                expression c
        function class Saturation_type_1() is Saturation
                expression c / (c + const(saturation_rate,0,0.1,10))
        function class Saturation_type_2() is Saturation
                expression c * c / (c * c + const(saturation_rate,0,0.1,10))
        function class Saturation_type_3() is Saturation
                expression (1 - exp(-const(saturation_rate,0,0.1,10) * c))
```

The Feeds_on process can be used for two different kinds of interactions that represent the predator dependence on several alternative food sources. The first interaction is conditional parallelism for food sources, i.e. when a population needs *each* of the several food sources at the same time. An example of such population is phytoplankton, which needs all of the essential inorganic nutrients at the same time. In the absence of one nutrient the growth will not occur. In this case a proper specification of this process for a phytoplankton (phyto) feeding on two nutrients, phosphorus (phosp) and nitrogen (nitro) will be **Feeds_on**(phyto,{phosp,nitro}). This process is transformed into a model in a following way (equation 9):

$$\text{Feeds\_on}(\text{phyto},\{\text{phosp},\text{nitro}\}) = \text{phyto} \cdot \text{Saturation}(\text{phosp}) \cdot \text{Saturation}\ (\text{nitro}) \tag{9}$$

The second interaction is when the predation happens as unconditioned parallel. This

means that although there might be a preference of the predator to some food source, it does not limit the growth of the predator if the predator has to use the second, les preferred source, instead. For example, suppose we want to model a zooplankton (zoo) feeding on two species of phytoplankton (phyto1 and phyto2). The task specification would be:

process Feeds_on(zoo,{phyto1}) feeds1
process Feeds_on(zoo,{phyto2}) feeds2

The transformation of this task into a model structure would combine the two processes additively in the differential equation (10) for zoo (recall the combining schemes in section 2.1.3).

$$zoo' = ... \text{ Feeds\_on}(zoo,\{phyto1\}) + \text{Feeds\_on}(zoo,\{phyto2\}) \quad (10)$$

The library can be used for inducing simple models in population dynamics domain. However, going into detailed analysis of model structures that can be derived from the library reveals some inconsistencies and incorrect model structures. The first is incorporation of temperature in the formulations of the ecological processes. According to the formalism temperature can be included in the Feeds_on process. For example, including a temperature influence in modelling of a phytoplankton consumption of two nutrients (n and p) requires the following specification in the task:

process Feeds_on(phyto,{n,p,temp})

Transformation into model gives the following formulation of the process:

Feeds_on(phyto,{p,n,temp}) = phyto*Saturation(p)*Saturation(n)*Saturation(temp)

where the temperature influence is formulated as:

Saturation(temp)=temp
Saturation(temp)=temp/(temp+const)
Saturation(temp)=temp*temp/(temp*temp+const)
Saturation(temp)=1- exp(-const*temp)

Except for the first formulation all of them are not consistent with our background knowledge. Temperature influence on ecological processes is commonly modelled in three general forms (1) linear response functions, (2) exponential response functions and (3) optimum temperature functions. Mathematical formulations of these functions can be found in e.g. Bowie et. al., (1985). Therefore, instead of the function *Saturation(temp)* another function is needed that would include all temperature influences. This requires a definition of the temperature influences as separate function

class in the knowledge library. Similarly, light intensity, which has great influence on primary producers should be defined as separate function class.

Another inconsistency can be found in the predator-prey interactions. This interaction between populations is handled with the Feeds_on process class as explained above. The process works correctly for the first conditioned parallel type of interactions- when all food items are needed at the same time. For example, the transformation of the process Feeds_on(phyto,{nitro,phosp}) gives correct model structures. Using the same process for the second unconditioned parallel type of interactions gives some incorrect model structures. For example, let's further develop the structure of the equation (10). Taking into account the transformation of the processes Feeds_on(zoo,{phyto1}) and Feeds_on(zoo,{phyto2}) into models we get:

$$zoo' = zoo \cdot Saturation(phyto1) + zoo \cdot Saturation(phyto2)$$

where Saturation(phyto1) and Saturation(phyto2) can have four possible formulations. Considering the second sub-class of the Saturation function class (Table 10) then zoo' becomes (11):

$$zoo' = k \cdot zoo \cdot \frac{phyto1}{k+phyto1} + k \cdot zoo \cdot \frac{phyto2}{k+phyto2} = zoo \cdot \left(\frac{phyto1}{k+phyto1} + \frac{phyto2}{k+phyto2} \right) \tag{11}$$

which is again incorrect model structure. The correct structure would be as shown in (12):

$$zoo' = k \cdot zoo \cdot \frac{phyto1+phyto2}{(phyto1+phyto2)+k} \tag{12}$$

The growth rate of zoo is limited with the sum of the food items that could be eaten by zoo. Thus new saturation function of form Saturation(phyto1+phyto2), or in general Saturation (food), where food is the sum of the food items, should be added in the taxonomy of function classes. This can not be incorporated in the existing Feeds_on process, i.e. this process can not be formulated so that it supports both feeding types. Either the process class supports the first or the later feeding type. Therefore it is reasonable to have two process classes – one for the first type of feeding which is typical for primary producers and another for the second type, common for secondary producers.

Note that correct model structure is obtained when the first sub-class of the Saturation function class is used. Then we have (13):

$$zoo' = const \cdot zoo^{*} phyto1 + const \cdot zoo \cdot phyto2 = const \cdot zoo \cdot (phyto1 + phyto2) \quad (13)$$

In some situations the process classes Growth and Feeds_on can both represent the same process, i.e. growth of a population, which can lead to incorrect model structures. For example, if we define in a same task specification that a population of a primary producer grows according *process Growth (phyto,{n,p,temp})* on one hand and at same time it feeds on nutrients using *process Feeds_on(phyto,{n,p})*, we can get an unusual model structure – addition of two growth processes that describe the same phenomena.

These inconsistencies with the background knowledge were eliminated in the new version of the knowledge library on food-web modelling in a lake (Atanasova et al., 2005). The most important differences are in the taxonomy of the process classes. In this library we have a growth process class (PP_Growth) that applies only on the primary producer population. The predator-prey interactions are handled with another process class, called Feeds_on. In this way any modelling task specification is transformed into a model structure that is consistent with the background knowledge. A scheme of the generalized knowledge included in the library is given Figure 4. The boxes represent the types of state variables, whereas the arrows stand for processes classes. The names of the physical and biochemical processes are given on the right hand side of the picture. As can be seen, many additional process classes are included so that the knowledge encoded in this library supports reconstruction of many well-known models of different complexity, such as the simple Vollenweider's model (Vollenweider, 1968) or the fairly complex SALMO (Bendorf, 1979; Recknagel, 1980). For details see Atanasova et al. (2005).

The main characteristics of the models that can be derived from this library are: 0-dimensional models, N-box models i.e., supports modelling of stratified lakes, fixed internal nutrient levels in primary producers and animals.

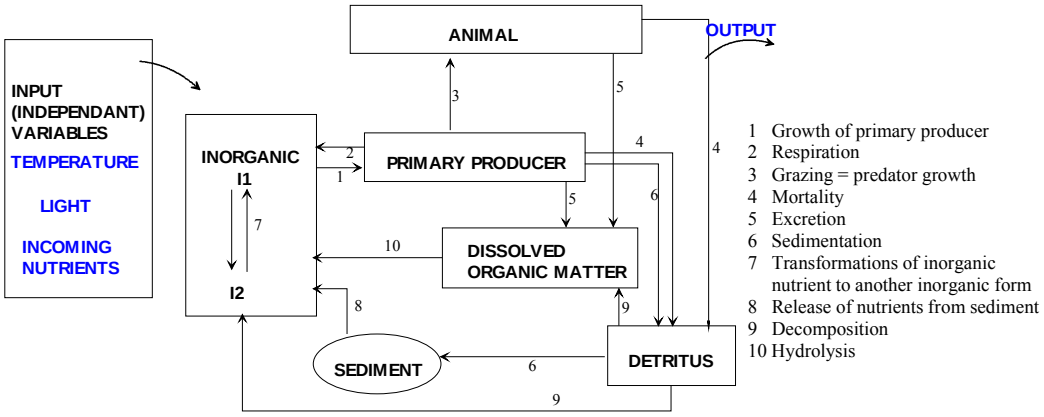


Figure 4: Generalized scheme of state variables (boxes) and interactions (arrows) in aquatic ecosystem

5. Evaluating the knowledge libraries on real life data

5.1 Lagoon of Venice

The Lagoon of Venice measures 550 km^2, but is very shallow, with an average depth of less than 1 m. It is heavily influenced by anthropogenic inflow of nutrients – 7 000 t/a of nitrogen and 1 400 t/a of phosphorus (Bendoricchio et al., 1994). These loads are highly above the Lagoon's admissible trophic limit and generate its dystrophic behaviour, which is characterized by excessive growth of algae, mainly macroalga *Ulva rigida*. This alga is not (notably) grazed by any animal or zooplankton, but shades-out the light by itself due to excessive growth in which also depletes all the nutrients. Inevitably, a massive die-out follows where all the oxygen is used for the decomposition of the dead algal biomass, which further drives the ecosystem to total crash-down of all higher organisms. During the decay phase, oxygen depletion, an irritating smell and disgusting look are threatening the tourism and fishery in the Venice lagoon. It is thus of utmost importance to understand this ecosystem and be able to predict future developments of the *Ulva rigida* growth.

Four sets of measured data were available (Coffaro et al., 1993). The data were sampled weekly for slightly more than one year at four different locations in the Lagoon. Location 0 was sampled in 1985/86, locations 1, 2, and 3 in 1990/91. The sampled quantities are nitrogen in ammonia (*nh*), nitrogen in nitrate (*no*), phosphorus in orthophosphate (*ps*) (all in μg/l), dissolved oxygen *DO* (in % of saturation), temperature *Temp* (°C), and algal biomass (*biomass)* (dry weight in g/m^2). In some experiments, we used the total nitrogen concentration *Ntot* instead of ammonia and nitrate nitrogen separately, as *Ulva* can use them both without greater difference as long as ammonia is not present in toxic concentrations (Coffaro et al., 1993; Bendoricchio et al., 1994).

Model induction from the simple library

Todorovski (2003) discovered a biomass equation (14) for Location 0 using the simple library explained in section 4.

$$\frac{dbiomass}{dt} = 6.17 \cdot 10^{-5} \cdot biomass \cdot (1 - \frac{biomass}{1.80}) + 3.01 \cdot 10^{-4} \cdot biomass \cdot DO \frac{no}{no + 6.28} - 0.0319 \cdot biomass \quad (14)$$

The simulations of this model show good fit to the measured data (Todorovski 2003). However, the structure of the model doesn't follow the background theoretical knowledge. Note that two terms (first and second term) are used to model the biomass growth process. The first is represents the logistic model for growth and the second represents limited growth model. As explained earlier this is not a common model formulation. All influences in the biomass growth process should be multiplicatively combined. Suppose there are no nutrients in the system. Then the second term would

correctly be equal to zero (no growth). But, according to this model (where the influences are sumarised) the biomass growth would still occur because of the first term Further, oxygen, which is actually a dependant variable (a consequence of biomass) plays a role of a nutrient for phytoplankton growth.

Another model (15), which quite accurately simulates the biomass values, was discovered for Location 2 (Todorovski 2003). Similarly as the model for Location 0 this model uses two terms for the same process (growth of biomass). The second term includes ammonia as the limiting nutrient and temperature influence, which is formulated inconsistently with the modelling knowledge (see section 4). Again, oxygen influence is formulated as if oxygen was a limiting nutrient.

$$\frac{dbiomass}{dt} = 4.79 \cdot 10^{-5} \cdot biomass \cdot (1 - \frac{biomass}{0.844}) + 0.406 \cdot biomass \cdot (1 - e^{-0.216 \cdot temp}) \cdot (1 - e^{-0.413 \cdot DO}) \cdot \frac{nh}{nh + 10} - 0.0343 \cdot biomass \quad (15)$$

Lagramge could not find any acceptable models for Locations 1 and 3.

Model induction from the complex library

We used more complex, newly developed knowledge library (Atanasova et al., 2005) to improve the above models. The expert knowledge was introduced in form of generic processes (see chapter 2.2), typical for algal dynamics. The processes considered in this case are growth, respiration, and mortality. The mortality process was introduced to account for self-shading and natural mortality, since there are no animals to graze on this alga. Because there were no data on light we introduced only nutrient and temperature influences on the growth process. Respiration and mortality were introduced as temperature dependant processes. This knowledge about the processes was introduced to Lagramge as shown in Table 11.

Table 11: Task specification for the Lagoon of Venice

1:	variable Inorganic po4
2:	variable Inorganic no3
3:	variable Inorganic nh3
4:	variable Primary_producer biomass
5:	variable Temperature temp
6:	process PP_growth(biomass, {po4,no3,nh3}, {temp}, {}) gr0
7:	process Respiration_PP(biomass, {temp},{}) resp0
8:	process Sedimentation(biomass) sed0
9:	process Mortality_PP(biomass, {},{temp},{}) mort

Measured variables in the system are declared in the lines from 1 to 5, i.e. *no3* (nitrate nitrogen), *nh3* (ammonia nitrogen), *po4* (dissolved inorganic phosphorus), *biomass* (macroalgae Ulva rigida), and *temp* (temperature). Processes are defined in the lines from 6 to 9. Biomass (macroalgae Ulva rigida) growth (PP_growth) is influenced by the inorganic nutrients and temperature. The third bracket {} is for light. Because it is left empty it indicates no known (measured) influence by light. The rest of the processes are respiration, sedimentation and mortality of biomass (Respiration_PP, Sedimentation and Mortality_PP).

Given the expert knowledge from Table 11 Lagramge discovered following biomass model (16) (with the smallest MDL error) on the data set from the measuring point 0:

$$\frac{dbiomass}{dt} = biomass \cdot 0.0522 \cdot \frac{ps}{ps+0} \cdot \frac{no}{no+3.26} \cdot (1-\exp(-8.7 \cdot nh)) \cdot \frac{temp}{7.8} - biomass \cdot 0.014 \cdot \frac{temp}{11.3}$$
$$-biomass \cdot 0.045 \cdot \frac{temp}{11.1} - biomass \cdot \frac{0.001}{1} \qquad (16)$$

The first term represents a biomass growth limited by the inorganic nutrients (po4, no3 and nh4), and temperature. As it is usual for marine lagoons, Lagramge found nitrogen as the limiting nutrient for algal growth. Note that the term $\frac{po4}{po4+0}$, which represents phosphorus limitation on growth is equal to 1, i.e. no limitation by phosphorus. Unlike the model discovered with the simple library (14) here nitrogen is used in it's both forms, i.e. nitrate and ammonium. Linear temperature response is applied to this process indicating optimal temperature around 19 ^{o}C. The other terms are respiration, with exponential temperature curve, simple mortality term and sedimentation, with settling coefficient of 0.001 m/day. The model performance is shown in Figure 5 (left).

In the next experiment Lagramge discovered a model for the measuring point 2 (17). The model is quite similar to the previous one except it takes the sum of nitrate and ammonia (n) as the limiting nutrient for growth, while respiration (second term) has more complex formulation. This formulation relates the algae respiration rate to the physiological conditions of the algal cells. It is a sum of two components: a low maintenance rate representing periods of minimal growth and a rate which is proportional to the maximum growth rate limited by the growth limitation factors (nutrients). Model performance is shown in Figure 5 (right).

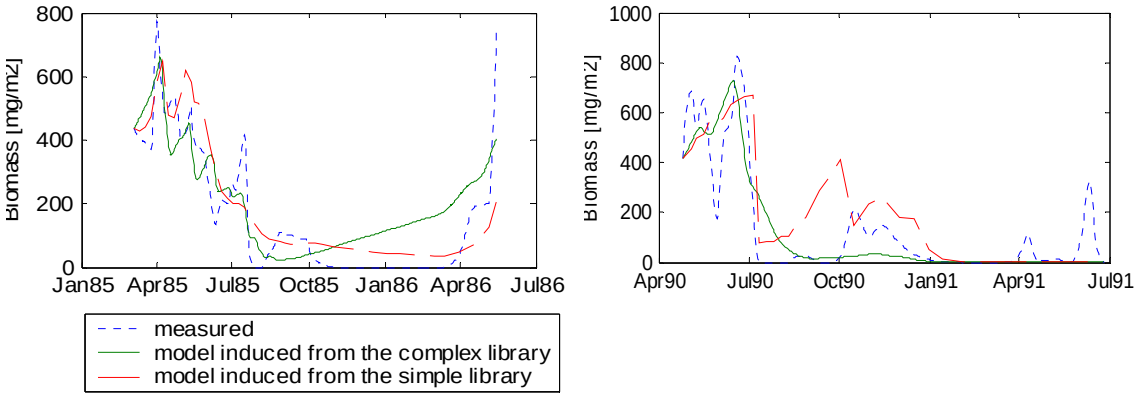


Figure 5: Simulation results of the biomass models, i.e. model induced from the complex library (solid line) and model induced from the simple library (dashed line). Left: models induced on the measuring point 0 data and right: models induced on measuring point 2 data.

$$\frac{dbiomass}{dt} = biomass \cdot 0.51 \cdot (1 - \exp(-2.6 \cdot n)) \cdot \frac{temp - 2.1}{19.99 - 2} - biomass \cdot 0.15 \cdot \left(\frac{temp - 4}{18 - 4} + 0.02 \cdot 1.13^{(temp-19)} \cdot \left(1 - \exp(-0.003 \cdot n)\right) \right) - biomass \cdot 0.19 \cdot 1.11^{(temp-20)} - biomass \cdot \frac{0.1}{1} \quad (17)$$

Using the simple library Lagramge couldn't find an acceptable model for locations 1 and 3 (Todorovski, 2003). Using the complex library Lagramge discovered a model for Location 3 (18) with performance as shown in Figure 6. Similarly as with the simple library Lagramge couldn't find acceptable model for point 1.

$$\frac{dbiomass}{dt} = biomass \cdot 0.13 \cdot \frac{ps}{ps + 0} \cdot (1 - \exp(-1.66 \cdot no)) \cdot (1 - \exp(-2.93 \cdot nh)) \cdot \frac{temp}{10.8} - biomass \cdot 0.016 \cdot 1.11^{(temp-17.4)} - biomass \cdot 1.3 \cdot \frac{temp}{6} - biomass \cdot \frac{0.001}{1} \quad (18)$$

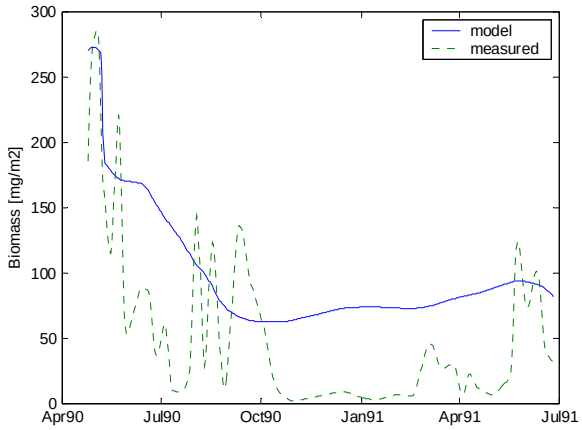


Figure 6: Performance of the Biomass model discovered on the measuring point 3 data

5.2 Lake Glumsoe

Lake Glumsø (Jørgensen et al., 1986) is situated in a sub-glacial valley in Denmark. It is shallow with average depth of about 2 m. Its surface area measures 266,000 m^2. For several years, it was receiving mechanically-biologically treated waste water from a community with 3,000 inhabitants and a surrounding area which was mainly agricultural with almost no industry. The high nitrogen and phosphorus concentration in the treated waste water has caused hypereutrophication. The lake contained no submerged vegetation, probably due to the low transparency of the water and oxygen deficit at the bottom of the lake.

The data set consists of 14 measurements in 2 months comprising daily through-flow, temperature, soluble nitrate (*no*), soluble phosphorus (*ps*), total phytoplankton biomass (*biomass*) and zooplankton biomass (*zoo*). The amount of measured data itself was far too small for automated modelling, so additional processing was applied to obtain a suitable data set (Kompare, 1995). Dotted graphs of the measurements were plotted and given to three human experts to draw a curve that, in their own opinion, described the dynamic behavior of the observed system variable between the measured points. A properly plotted expert curve can be regarded as an additional source of reliable data. Curves drawn by human experts were then smoothed with Besier splines.

Todorovski (2003) discovered a model (19) using the measurements and the modelling knowledge from the simple library.

$$\frac{\partial biomass}{\partial t} = biomass \cdot 0.553 \cdot \frac{ps}{0.084 + ps} \cdot temp - 4.35 \cdot biomass - 8.67 \cdot zoo \cdot biomass \tag{19}$$

The model structure is consistent with the expert knowledge. It has three terms

representing three processes (1) phytoplankton growth, (2) loss (decay) of phytoplankton and (3) loss due to grazing by zooplankton. Growth of the phytoplankton population is nutrient limited, where phosphorus (ps) is found to be the limiting nutrient. Temperature influence is incorporated in the process formulation. The grazing (predation) process is formulated in accordance with the Voltera-Lotka model, i.e. the predation rate is proportional to the densities of the predator (zoo) and the pray (phyto) populations.

Slightly different model (20) was discovered using the complex library (Atanasova, 2004).

$$\frac{dbiomass}{dt} = biomass \cdot 1.5 \cdot (1 - e^{-2.52 \cdot ps}) \cdot \frac{ns}{0 + ns} \cdot \frac{temp - 4}{15.4 - 3} - 0.5 \cdot biomass - 9.66 \cdot zoo \cdot \frac{temp}{4.2} \cdot (1 - e^{-0.12 \cdot biomass}) \quad (20)$$

Again, phosphorus was found to be the limiting nutrient for phytoplankton growth. The grazing process was formulated differently from the previous model. Here, the process is influenced by temperature and phytoplankton is included as food limiting factor for zooplankton growth. The performance of the both models is shown in Figure 7. Compared with the measurements, the model discovered using the complex library shows a slightly better performance then the other one.

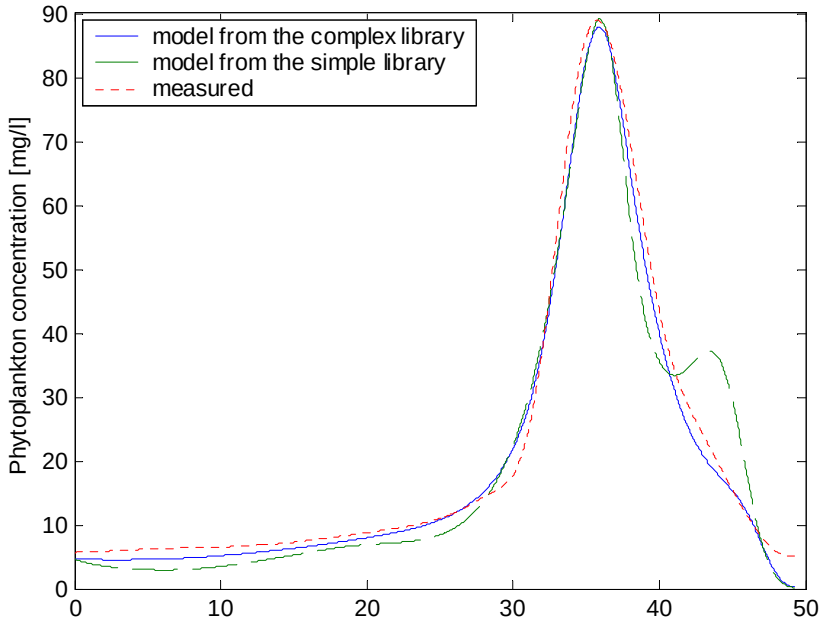


Figure 7: Performance of the models discovered using the simple library (Todorovski, 2003) (dashed line) and using the complex library (Atanasova et. al, 2004) (solid line)

6. Conclusions

An analysis of the background knowledge incorporation into automated modelling framework, i.e. into ODE mathematical model induction from data has been performed. In particular we analysed two knowledge libraries: (1) a simple library developed mainly for illustration of the automated modelling method (Todorovski, 2003) and (2) complex library containing great part of the lake modelling knowledge (Atanasova et al., 2005). Both libraries were tested on real world data –the Lagoon of Venice, Italy and the lake Glumsoe, Denmark. The background knowledge has great influence on the resulting models, obtained by the automated modelling procedure. In the case of the Lagoon of Venice we obtained better model structures for two measuring locations with the complex library, whereas the accuracy was similar to the models obtained with the simple library. Further, using complex library Lagramge discovered another model with satisfactory accuracy for Location 3. A model for this location could not have been discovered with the simple library (Todorovski, 2003). For the lake Glumsoe a model of correct structure was obtained with the simple library (Todorovski, 2003). Using the complex library Lagramge found slightly different model structure, which also performs better regarding the measured data.

References

Atanasova, N., Todorovski, L., Džeroski, S. and Kompare, B. (2005): Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *Ecological Modelling*.

Bendorf, J. 1979. A contribution to the phosphorus loading concept. *Int. Revue ges. Hydrobiol.*, 64(2): 177-188.

Bendoricchio, G., Coffaro, G. & De Marchi, C. 1994. A trophic model for Ulva rigida in the Lagoon of Venice. *Ecological Modelling*, 75-76: 485-496.

Bowie, G.L., Mills, W.B., Porcella, D.B., Campbell, C.L., Pagenkopf, J.R., Rupp, G.L., Johnson, K.M., Chan, P.W.H., Gherini, S.A. & Chamberlin, C.E. 1985. Rates Constants and Kinetic Formulations in Surface Water quality Modelling. Athens, GA: US EPA, ORD.

Chapra, S.C. 1997. *Surface Water-Quality Modeling*: McGraw-Hill.

Coffaro, G., Carrer, G. & Bendoricchio, G. 1993. Model for Ulva Rigida growth in the Lagoon of Venice., *UNESCO MURST Project: Venice Lagoon Ecosystem*. Padova, Italy: University of Padova.

DeAngelis, D.L. 1992. *Dynamics of Nutrient Cycling and Food Webs*. London: Chapmann & Hall.

Joergensen, S.E. & Bendoricchio, G. 2001. *Fundamentals of Ecological Modelling, Third Ed*. Amsterdam: Elsevier Science Ltd.

Kompare, B. 1995. *The Use of Artificial Intelligence in Ecological Modelling*. Ph.D. Thesis, FGG, Ljubljana; Royal Danish School of Pharmacy, Copenhagen, Ljubljana, Copenhagen.

Kompare, B. & Džeroski, S. 1995. *Getting more out of data: Automated modelling of algal growth with machine learning*. Paper presented at the the International symposium on coastal ocean space utilisation, Yokohama, Japan.

Langley, P., Sanchez, J., Todorovski, L. & Dzeroski, S. 2002. *Inducing process models from continuous data*. Paper presented at the The Nineteenth International Conference on Machine Learning, Sydney Australia.

Recknagel, F. 1980. *Systemtechnische Prozedur zur Modellierung und Simulation von Eutrophierungs-prozessen in stehenden und gestauten Gewässern*. , TU Dresden, Dresden.

Todorovski, L. & Dzeroski, S. 2001. *Using Domain Knowledge on Population Dynamics Modeling for Equation Discovery*. Paper presented at the Proceedings of the Twelth European Conference on Machine Learning, Freiburg, Germany.

Todorovski, L. 2003. *Using Domain Knowledge for Automated Modeling of Dynamic Systems with Equation Discovery*. , University of Ljubljana, Ljubljana, Slovenia.

Vollenweider, R.A. 1968. The scientific basis of lake and stream eutrophication with particular reference to phosphorus and nitrogen as eutrophication factors. Paris: Organisation for Economic Cooperation and Development.

20. Computational Assemblage of Ordinary Differential Equations for Chlorophyll-*a* Using a Lake Process Equation Library and Measured Data of Lake Kasumigaura

N. Atanasova · F. Recknagel · L. Todorovski · S. Džeroski · B. Kompare

20.1 Introduction

Lake ecosystems are highly complex dynamic systems. Modelling of such ecosystems is an ongoing challenge to scientists, who continue to gain better understanding of ecological processes in order to more realistic simulate ecosystem behaviours. Two basic modelling approaches can be distinguished: the deductive, knowledge driven approach resulting in deterministic models, and the inductive, data driven approach exploring candidate models and match them with measured data resulting in empirical models.

Deterministic models are typically represented by ordinary differential equations (ODE) which are being applied to lake ecosystems since the 1970s (e.g. Straskraba and Gnauck 1984; Recknagel 1989; De Angelis 1992; Chapra 1997; Jorgensen and Bendoricchio 2001). If applied to real lake data ODE can be well adjusted and interpreted in the context of the domain due to their explicit causality. However, complex ecological processes are often not yet fully understood and therefore ODE are sometimes adapted to our incomplete knowledge resulting in simplified models.

By contrast inductive models induced from the data by bio-inspired computation such as artificial neutral networks and evolutionary algorithms may rely heavily on the comprehensiveness of data. They have been demonstrated to be powerful predictive tools (e.g. Recknagel et al. 2002; Lee et al. 2005) but may still be limited in their representation and explanation.

In this paper we apply an approach that combines both domain knowledge and data. The domain knowledge is gathered in a knowledge library, which is used to guide the process of induction from real data. The result is a set of elementary process descriptions for ODE that match basic principles of the domain of interest (Todorovski and Dzeroski, 2001; Langley et Al., 2002; Todorovski, 2003). In the early days of the development of these tools (Todorovski & Džeroski, 1997), the knowledge had to be provided as an explicit definition of the space of candidate models. Now, these tools allow the user to provide higher-level domain knowledge about building mathematical models of complex real-world systems.

In this paper we apply the combined modelling approach to Lake Kasumigaura (Japan) by utilising a library for process equations of lake domain knowledge and measured data. Previous research on modelling of lake Kasumigaura was based on artificial neural networks (ANN), genetic algorithms (GA) and evolutionary algorithms (EA). ANN was trained to predict the dominant algal genera (Recknagel et al. 1997; Recknagel et al. 1998; Wei et. al., 2001) and zooplankton abundance (Recknagel et al. 1998) in Lake Kasumigaura. GA was applied to induce predictive ODE for Chl-a (Whigham and Recknagel 2001) and EA to induce predictive rules for Chl-a in the lake (Bobbin and Recknagel 2001; Recknagel et al. 2002). In the context of this research we attempt to discover predictive ODE for the Chl-a by assembling and adapting process equations from a lake domain library.

20.2 Methods and Material

20.2.1 LAGRAMGE: Computational Assemblage of ODE

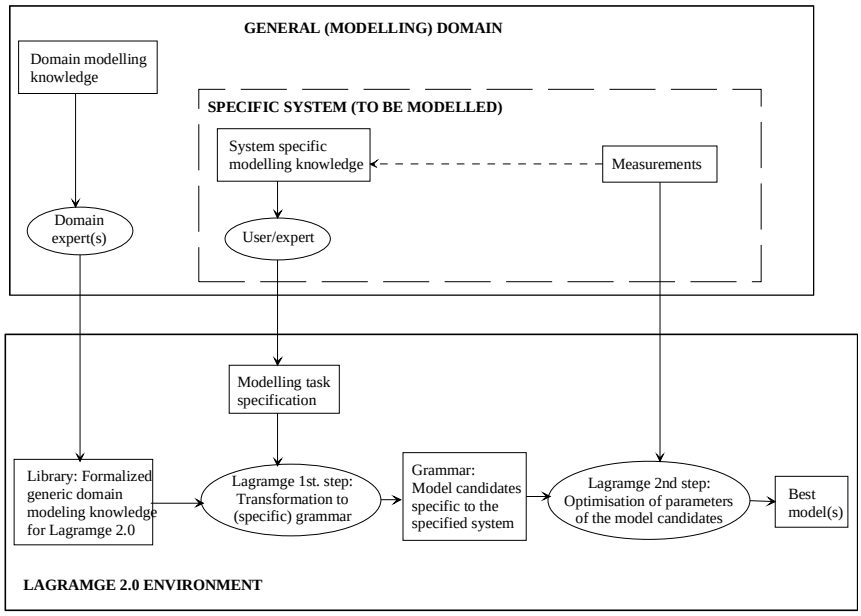


Fig. 20.1. An automated modeling framework based on the integration of domain-specific modeling knowledge in the process of equation discovery

The principal concept of computational assemblage of ODE by LAGRAMGE is shown in Fig. 20.1. After the modelling task has been defined and the lake data been specified domain knowledge is transformed from the library into a grammar. This grammar specifies the space of candidate models as illustrated in the left part side of Fig. 20.1. Once the grammar has been determined, LAGRAMGE is heuristically searching through the space of candidate models and testing each of them with measured data after fitting constant parameter values. These models are evaluated by means of two error measurements, i.e. mean square error (MSE) and MDL computed by LAGRAMGE. Further details about the algorithm of LAGRAMGE can be found in (Todorovski, 2003).

20.2.1 Domain Knowledge Library for Lake Ecosystems

In order to be used in the model induction procedure, the knowledge needs to be coded in the knowledge library. Todorovski (2003) developed the formalism for encoding the domain knowledge about lake ecosystems. Using this formalism Atanasova et al. (2004) developed a comprehensive knowledge library for lakes ecosystems. The library supports the construction of 0-dimensional N-box models, i.e., supports modelling of stratified lakes. The equations coded in the library are recruited from literature models developed for lakes, and can be assembled to different levels of ecosystem structures such as the simple Vollenweider model (Vollenweider, 1968) or the fairly complex model SALMO (Benndorf and Recknagel 1982; Recknagel and Benndorf 1982). For more details see Atanasova (2004).

In general, the knowledge coded in the library can be conceptually presented as shown in Fig. 20.2, where only a part of the library is depicted. The boxes represent the types of state variables, whereas the arrows stand for ecological processes that influence the state variables. According to this diagram the library allows for modelling of dissolved inorganic nutrients (e.g. inorganic nitrogen, phosphorus and silica), primary producers (e.g. diatoms and green algae), secondary producers (e.g. zooplankton), dissolved organic matter and detritus. Processes, which are in the library, but not depicted on Fig. 20.2 are describing dissolved oxygen pathways such as aeration, oxygen production or consumption processes.

The knowledge in the library is formalized in terms of the: (1) taxonomy of variable types, (2) taxonomy of basic processes that govern the behavior of the state variables, (3) alternative models of the basic processes, and (4) knowledge how to combine models of individual processes to a system of ODE for an ecosystem.

Basic processes (arrows in Fig. 20.2) are declared as *process classes*. A process class represents different formulations of a certain basic process. For example, the process that describes a primary producer growth (arrow no. 1 in Fig. 20.2) includes the exponential, logistic and limited growth models. Furthermore, the limited growth model includes different formulations growth functions limited by

nutrients, light or temperature.

According to the ODE for the state variables the classes of processes are combined by so-called *combining schemes*. Combining scheme of specific variable represent the all processes that may affect that variable. In other words, each combining scheme represents a differential equation for the specific state variable. Thus, the library contains six combining schemes for six dependant (state) variable types.

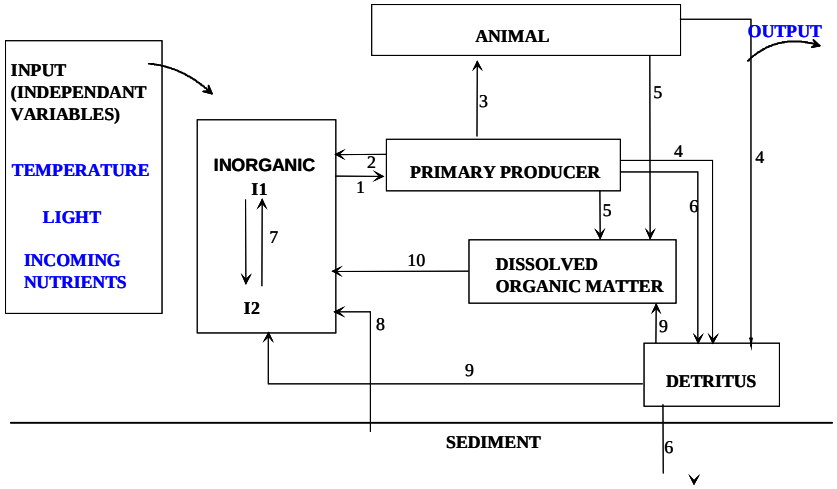


Fig. 20.2. Generalized scheme of compartments and interactions

20.2.2 Task Specification

The domain knowledge library comprises general knowledge about modelling of lakes. In the task specification the user of LAGRAMGE provides the specific knowledge and data of the lake to be modelled. It includes the selection of variables and processes. The model variables are specified by the variable type and the variable name as follows:

variable variable_type 'variable_name'

The word **system** in front of the word **variable** variable specifies a state variable.

The process variables are defined by the word ***process***, followed by the process name and the process arguments:

process 'process name'(argument1, argument2, ...) **process_notation**

Arguments represent the variables in the observed system that influence (or are influenced by) the specific process. They are used in the process formulations in the library. If some of the arguments in the process are considered as sets within the process then we put the names of those arguments into brackets {}. A set can contain none (empty), one or many variables (arguments) of the same type.

Tab. 20.1. Declared variable types in the knowledge library

Variable type	Description	dependant (state) / independent (forcing)
type Concentration **is real**	concentration of a substance	generic
type Light **is real**	light intensity	independent
type Temperature **is real**	temperature	independent
type Precipitation **is real**	precipitations	independent
type Flow **is real**	flow rate	independent
type Area **is real**	contributing area of the incoming nutrients	independent
type Inorganic **is Concentration**	dissolved inorganic nutrients	dependant
type Population **is Concentration**	concentration of a population	generic
type Detritus **is Population**	particulate dead organic matter	dependant
type Oxygen **is Concentration**	dissolved oxygen	dependent
type Dom **is Concentration**	dissolved organic matter	dependant
type Primary_producer **is Population**	primary producers	dependant
type Animal **is Population**	secondary producers	dependant

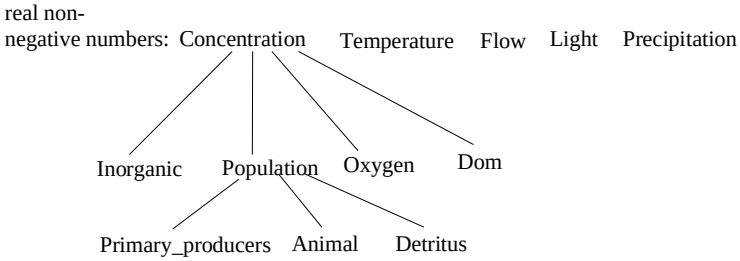


Fig. 20.3. Graphical presentation of variable types and sub-types in the knowledge library for lake modelling

Thus, in order to correctly introduce the expert knowledge to LAGRAMGE we need to know the: (1) types of variables declared in the library and (2) types of ecological processes declared in the knowledge library. The types of the variables in the knowledge library for lakes are given in Tab. 20.1. The type *Concentration*

is a generic variable type that is determined by sub-types of variables. It has four sub-types, i.e. *Inorganic* representing the dissolved inorganic nutrients, *Population* representing particulate organic matter, *Dom* denoting a dissolved organic matter and *Oxygen* representing dissolved oxygen concentration. Population has again three sub-types – *Primary_producers*, *Animal* and *Detritus*. The types of variables are schematically shown in Fig. 20.3.

If we want to model interactions between several species (for example primary producer grazing on more then one nutrient) we need to declare sets of variables. Declaration of set Primary_producers of the type Primary producer is given below. Please note the plural form of the set name, which is derived from the singular name of the variable type name:

type Primary_producers **is set(**Primary_producer**).**

Tab. 20.2. Description of process' definition in the knowledge library

	Process description	Process name	Arguments: types of variables involved in the process' formulations	Argument declared as Set: y/n
1	Outflow of a substance from the system	Outflow	1. Concentration 2. Flow	n n
2	Inflow of a substance to the system	Inflow	1. Concentration 2. Concentration 3. Flow	n n n
3	Settling of a substance	Sedimentation	1. Concentration 2. Temperature	n y
4	Diffusion	Diffusion	1. Concentration 2. Concentration	n n
5	Growth of a primary producer	PP_growth	1. Primary_producer 2. Inorganic 3. Temperature 4. Light	n y y y
6	Predator prey interactions	Feeds_on	1. Animal 2. Population 3. Temperature	n y y
7	Respiration of a primary producer	Respiration_PP	1. Primary_producer 2. Inorganics 3. Temperature 4. Light	n y y y
8	Respiration of an animal (sec. prod)	Respiration_A	1. Animal 2. Temperature	n y
9	Natural mortality of a primary producer	Mortality_PP	1. Primary_producer 2. Inorganic 3. Temperature 4. Light	n y y y
10	Natural mortality of an animal (sec. prod)	Mortality_A	1. Animal 2. Temperature	n y
11	Excretion from secondary producers	Excretion_A	1. Animal 2. Temperature	n y

The Tab. 20.2 includes the description of the majority of the processes declared

in the knowledge library. In the first column the description of the ecological processes is given. The second column contains the processes' names as they are declared in the library. The third and the fourth column give information about the arguments, i.e. the variables involved in the processes' formulations. In the third column the types of the involved variables (arguments) are listed, whereas the fourth contains information whether the variable is included in the process declaration as set or not.

For example, in line 5 the definition of the ecological process Growth of a primary producer is given. The process name is PP_growth and it has 4 arguments. The first is of type Primary_producer and it represents the variable which the process refers to. The rest of the arguments are variables of types Inorganic, Temperature and Light. They are all declared in the library as sets. The statement in the task specification ***process*** PP_growth(phyto1, {ps}, {temp}, {llight}) **growth,** describes the growth of a primary producer *phyto1*. The process is influenced by single inorganic nutrient *ps*, temperature *temp* and light *light* respectively. Living one of the brackets {} empty would indicate no influence by the variable which was left out. For instance definition of a growth process of phytoplankton (phyto1) that is influenced by two nutrients phosphorus (ps) and nitrogen (ns) and temperature (temp), but not light (light) limited would be:

process PP_growth(phyto1, {ps, ns}, {temp}, {}) **growth.**

Note that this specific "Lake" knowledge library includes several formulations for each of the process classes in the task specification (Atanasova et al., 2004). For example the process class PP_growth contains five different models for primary producer growth, i.e. exponential, logistic, growth limited by temperature, light and nutrients, growth limited model that accounts for variable optimal temperature and growth limited model that couples the effects of light and temperature. Furthermore, light, temperature and nutrients limitations are defined as function classes that include several different formulations for each. Thus, we have more than fifty possible formulations for the PP_growth process, which are all correct from the standpoint of the used library and defined task. Similarly, we have several possible formulations for the rest of the process classes in this system.

In order to find a model of a specific system with Lagramge we need (1) measurements of the state (dependent) and forcing (independent) variables that will be used in the optimisation procedure and (2) expert knowledge about the variables and processes, which will be used for determining the model structure.

20.2.3 Data of Lake Kasumigaura

Lake Kasumigaura is a shallow lake in Japan with maximal depth of 7 m and average depth of 4 m. It has a volume of 662 million m^3 and a surface area of 220 km^2. The hypereutrophic state of the lake causes blue-green algal blooms in summer and autumn with frequently high abundances of *Microcystis* and *Oscillatoria.* The Tab. 20.3 summarises the measured data of Lake Kasumigaura

from 1986 to 1992 that were used as in a daily interpolated format in the context of this study.

Tab. 20.3. Structure of the database of Lake Kasumigaura for 1986 to 1992

Limnological Variables	**Mean / Min / Max**
PO_4 µg/l	14.16 / 1 / 235
NO_3 mg/l	0.52 / 0.001 / 2.39
Si mg/l	3.29 / 0.015 / 12.49
Chla µg/l	74.5 / 0.69 / 279.5
Water Temperature ^oC (WT)	16.37 / 2.1 / 32
Solar Radiation Jcm^{-2} day^{-1}	1281 / 65 / 3364
Phytoplankton cells/ml *Microcystis* and *Oscillatoria* *Scenedesmus* *Synedra* Zooplankton individuals/l *Cladocera*	

20.2.4 Experimental Framework

In order to test the performance of the LAGRAMGE algorithm for the simulation of chlorophyll-a (chl-*a*) by means of ODE assembled and adapted to data from Lake Kasumigaura following experiments were designed and conducted:

- Experiment 1: Discover chl-*a* models for each year separately. This experiment focused on the question whether it is possible to find a generic model structure for all years from 1986 to 1992 and just optimise the parameter values for each year or to require specific model structures for each year. We tested each year-specific model on the remaining years in order to find out whether there is a generic model for all measured years. Algal grazing by zooplankton was not included in this experiment as zooplankton data were only available for the years 1986 to 1989.
- Experiment 2: Discover one chl-*a* model for all years from 1986 to 1992. This experiment focused on the question whether it is possible to derive a generic model from all data that would be valid for each single year. The model was trained by data from 1986 to 1991, and tested for the year 1992. Algal grazing by zooplankton was not included in this experiment as zooplankton data were only available for the years 1986 to 1989.
- Experiment 3: Discover one chl-*a* model including algal grazing by zooplankton by using the years 1986 to 1988 for learning and 1989 for testing.

The task specification for experiment (3) is given in Tab. 20.4. Following types of variables are declared: inorganic nutrients, i.e nitrogen_nitrate (*no3*), dissolved inorganic phosphorus (*ps*) and silica (*silica*), primary producer (*chla*), animal (*clad*), temperature (*temp*) and light (*light*). The word **system** in front of the primary producer declaration denotes that only *chla* model will be discovered

(*chla* is the only state variable), while the rest of the variables will be considered as independent variables. The processes are declared in lines from 8 to 11. Phytoplankton growth is described in line 8 (recall the process description from the previous section). The process Feeds_on (line 9) stands for (1) predatory loss of phytoplankton (chla) and (2) growth of zooplankton (clad). Optional arguments of this process are the food (phyto) and temperature (temp), which means that the growth of clad can be or not influenced by the food (none or many species) and temperature. Similarly the rest of the processes in the system (Respiration_PP, and Sedimentation) are defined (see lines 10 and 11).

Tab. 20.4. Modelling task specification for lake Kasumigaura

1:	variable Inorganic ps
2:	variable Inorganic no3
3:	variable Inorganic silica
4:	system variable Primary_producer chla
5:	variable Animal clad
6:	variable Temperature temp
7:	variable Light light
8:	process PP_growth(chla, {ps, no3, silica}, {temp}, {light}) gr1
9:	process Feeds_on(clad, {chla}, {temp}) feeds1
10:	process Respiration_PP(chla, {temp},{},{}) resp1
11:	process Sedimentation(chla, {temp}) sed1

According to the experimental setup the grazing process (Feeds_on) was either included or excluded from the induction procedure. The task specification from Tab. 20.4 was modified for this case by replacing the process Feeds_on by natural mortality (Mortality_PP): process Mortality_PP(chla, {temp}, {}, {}) mort1.

According to the combining schemes (mass balances) declared in the library, this task specification gives either the model structure as (20.1), or in the case of replacing the Feeds_on process (predatory loss) by natural mortality as (20.2):

$$\frac{dchla}{dt} = \text{PP\_growth - Respiration - Sedimentation - Feeds\_on} \qquad (20.1)$$

$$\frac{dchla}{dt} = \text{PP\_growth - Respiration\_PP - Mortality\_PP - Sedimentation} \qquad (20.2)$$

Note that the formulation of the process loss of phytoplankton by grazing needed some adjustments since the zooplankton abundance unit [individuals/l] was not compatible with the biomass unit [mass/volume]. We overcame this problem by allowing only one possible formulation of the Feeds_on process in the knowledge library, i.e:

$$\text{Feeds\_on (Grazing)} = C_{f\max} \cdot f1(temp) \cdot f2(F_T) \cdot clad \cdot chl\_a$$

where C_f is zooplankton filtration rate [ml/(individuals*time)], *clad* is the abundance of cladocera in [individuals/ml], *chl_a* is chlorophyll-*a*

concentration in [mg/l chl-*a*], f1(temp) is temperature influence function (unitless) and f(F_T) is food limitation function for zooplankton growth (unitless). In this case F_T represents the total phytoplankton concentration. Considering this, the loss of phytoplankton is calculated in [mg/l chl-*a*].

20.3 Results and Discussion

20.3.1 Experiment 1

This experiment aimed to identify separate ODE models for the calculation of chl-*a* for each years. Thus the LAGRAMGE algorithm discovered 7 models with corresponding MSE and MDL function. The models with the minimal MDL values for each year were chosen as best models, i.e. equation (20.3) was the best model for 1986, equation (20.4) for 1987, equation (20.5) for 1988, equation (20.6) for 1989, equation (20.7) for 1990, equation (20.8) for 1991 and equation (20.9) for 1992:

$$\frac{dchla}{dt} = chla \cdot 0.152 \cdot \frac{ps}{ps+0} \cdot \frac{no3^2}{no3^2+4.7E-7} \cdot \frac{silica^2}{silica^2+0.011} \cdot \frac{temp-0}{15-5} \cdot \frac{light}{light+196.7} - chla \cdot 0.1 \cdot \frac{temp-5}{17.4-2.5} - \\ - chla \cdot chla \cdot 0.001 - chla \cdot \frac{0.04}{5} \tag{20.3}$$

$$\frac{dchla}{dt} = chla \cdot 0.08 \cdot \frac{ps^2}{ps^2+3.2E-6} \cdot \frac{no3}{no3+0.00012} \cdot \frac{silica^2}{silica^2+0.023} \cdot \frac{temp}{16.2} \cdot \frac{light}{light+41.8} - chla \cdot 0.005 \\ - chla \cdot 0.01 \frac{temp-0}{15-5} - chla \cdot \frac{0.096}{5} \cdot 1.11^{(tmp-15)} \tag{20.4}$$

$$\frac{dchla}{dt} = chla \cdot 0.09 \cdot \frac{ps}{ps+0} \cdot \frac{no3}{no3+0} \cdot \frac{silica}{silica+0.022} \cdot \frac{temp}{10.8} \cdot \frac{light}{light+200} - chla \cdot 0.022 \cdot 1.11^{(temp-18.8)} - \\ - chla \cdot 0.01 \cdot \frac{temp}{7.2} - chla \cdot \frac{0.05}{5} \tag{20.5}$$

$$\frac{dchla}{dt} = chla \cdot 0.09 \cdot \frac{ps}{ps+0} \cdot \frac{no3}{no3+0} \cdot \frac{silica}{silica+0} \cdot \frac{temp}{6.4} \cdot \frac{light}{light+200} - chla \cdot 0.02 \cdot 1.13^{(temp-15)} - \\ chla \cdot chla \cdot 0.77 - chla \cdot \frac{0.14}{5} \tag{20.6}$$

$$\frac{dchla}{dt} = chla \cdot 0.134 \cdot \frac{ps}{ps + 3.2E-5} \cdot \frac{no3}{no3 + 0} \cdot \frac{silica}{silica + 0} \cdot \frac{temp}{19.8} \cdot \frac{light}{light + 0} - chla \cdot 0.004 \cdot 1.12^{(temp-20)} - chla \cdot chla \cdot 0.54 - chla \cdot \frac{0.28}{5} \cdot \frac{temp-5}{15-5} \tag{20.7}$$

$$\frac{dchla}{dt} = chla \cdot 0.224 \cdot \frac{ps}{ps + 0} \cdot \frac{no3}{no3 + 0} \cdot \frac{silica}{silica + 0} \cdot \frac{temp}{20} \cdot \frac{light}{light + 10.3} - chla \cdot 0.0009 - chla \cdot chla \cdot 0.332 - chla \cdot \frac{0.5}{5} \cdot \frac{temp-2}{15-5} \tag{20.8}$$

$$\frac{dchla}{dt} = chla \cdot 0.139 \cdot \frac{ps}{ps + 0} \cdot \frac{no3}{no3 + 0} \cdot \frac{silica}{silica + 0} \cdot 1.11^{(temp-19)} \cdot light \cdot e^{-\frac{\left(\frac{light}{179.5}+1\right)}{184.55}} - chla \cdot 0.056 \cdot 1.12^{(temp-15)} - chla \cdot chla \cdot 0.023 \cdot \frac{temp}{1.3} - chla \cdot \frac{0.0001}{5} \tag{20.9}$$

The alternative model structures include processes as shown in equation (20.2). In all cases the growth term is dependent on nutrient concentrations, water temperature and underwater light. Nutrient limitation functions for ps, no3 and silica are formulated with the two variations of Monod term, i.e. $f(x) = \frac{x}{x + \text{constant}}$ or $f(x) = \frac{x^2}{x^2 + \text{constant}}$. Note that the smaller the constant (half saturation coefficient) in the Monod term the smaller is the influence by *x*. For example, a term with saturation coefficient zero, i.e., $\frac{x}{x+0}$ is equal to 1, which means no limitation (influence) by *x*. From this we can reveal the nutrients' influence on the total phytoplankton growth and how the limiting nutrient(s) is changing with time. According to the models this influence is pretty unpredictable, which is probably a result of the variety of algae species, in the total phytoplankton. Phosphorus was found to be the limiting nutrient only in 1990, and in 1987 together with nitrate and silica. Also the saturation constant in the phosphorus limitation function is very small. Nitrogen was the limiting nutrient in 1986 (together with silica) and 1987 (together with phosphorus and silica), while silica was limiting in 1986, 1987 and 1988. Nutrients did not limit the phytoplankton growth in 1989, 1991 and 1992. To find the limiting nutrient it is crucial (1) to know the load of the lake with the nutrients (external and internal) and (2) to estimate which algae will bloom most severely. The latest is partly revealed by the models. In 1986 nitrogen limitation can be related with the severe microcistis blooms. There were small amounts of diatoms, obviously limited by silica. It is surprisingly for the 1987 model that all nutrients were found as limiting, although there are no diatoms identified in this year. Severe blooms of diatoms in 1988 were limited by silica as revealed by equation (5). According to the discovered models the lake receives quite a lot of nutrients, since the nutrients were not limiting the growth in 1989 1991 and 1992, and in 1990 the limitation by phosphorus is negligible.

Monod expression is used for light limitation function in all models except for 1992, where the photoinhibition formulation for light is used. Temperature influence is modelled with the linear temperature curve in all years except for 1992, when the influence is exponential. The rest of the processes, i.e. respiration, mortality and sedimentation are modelled with similar formulations in all models. The models differ greatly in the parameter values that may suggest that some of them should be replaced by variables.

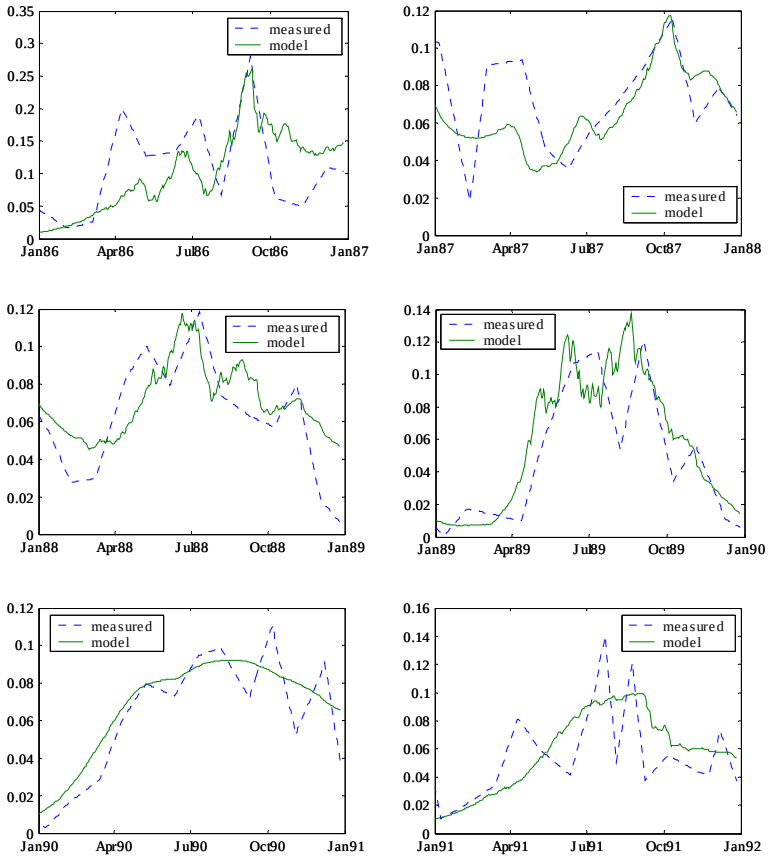


Fig. 20.4. Simulation results of the chl-*a* [mg/l] equations (20.3) to (20.9) assembled and trained by Lake Kasumigaura data of 1986 to 1992

The performance of the 6 models compared to the measured data is shown in Fig. 20.4. Most of the models are able to approximate well the timing and magnitude of chl-*a*.

In order to find a model that would simulate the phytoplankton during the entire period satisfactorily, each model was validated on the data set that was not used for training of that specific model. None of the discovered models could accurately simulate chl-*a* on unseen data, except for the equation (20.5) discovered for the data of 1988. The validation of this model is shown in Fig. 20.5. The model performs satisfactorily, except in 1986. This year seems to be quite unusual, since the chl-*a* peak is nearly twice as much as it is in the rest of the years.

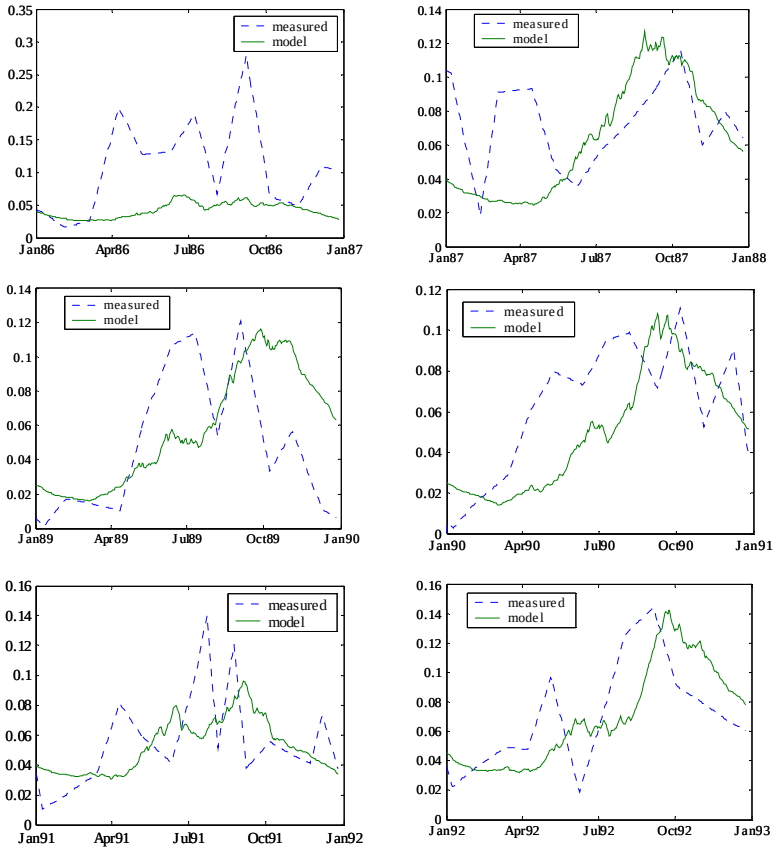


Fig. 20.5. Simulation results of the chl-*a* [mg/l] equation (20.5) assembled and trained by Lake Kasumigaura data of 1988 and tested by the data of 1986, 1987, 1989 to 1992

20.3.2 Experiment 2

The first experiment has clearly demonstrated the highly dynamic nature of algal biomass represented in Lake Kasumigaura reflected by the calculated data of 6 annually specific ODE and the measured data for chl-*a*. Both vary distinctly in timing and magnitude year by year. To find a generic model structure that would accurately simulate chl-*a* dynamics for consecutive years is therefore very challenging. However the equation (20.5) discovered for the data of 1988 in the experiment 1 has indicated that LAGRAMGE can discover common patterns in complex data, and that the year 1988 provides average lake data which are suitable for training the chl-*a* model. Our second experiment aimed at the discovery of a generic chl-*a* model trained by data of all years 1986 to 1991. The ODE structure was specified according to equation (20.2). The ODE for chl-*a* with the lowest MDL identified by LAGRAMGE reads sa follows:

$$\frac{dchla}{dt} = chla \cdot 0.117 \cdot \frac{ps^2}{ps^2 + 2.6E-07} \cdot \frac{no3}{no3 + 9.8E-05} \cdot \frac{silica}{silica + 0} \cdot \frac{temp}{20} \cdot \frac{light}{light + 200} - chla \cdot 0.00658$$
$$- chla \cdot 0.003 \cdot \frac{temp}{3.3} - chla \cdot \frac{0.072}{5} \cdot 1.1^{(temp-18.1)} \qquad (20.10)$$

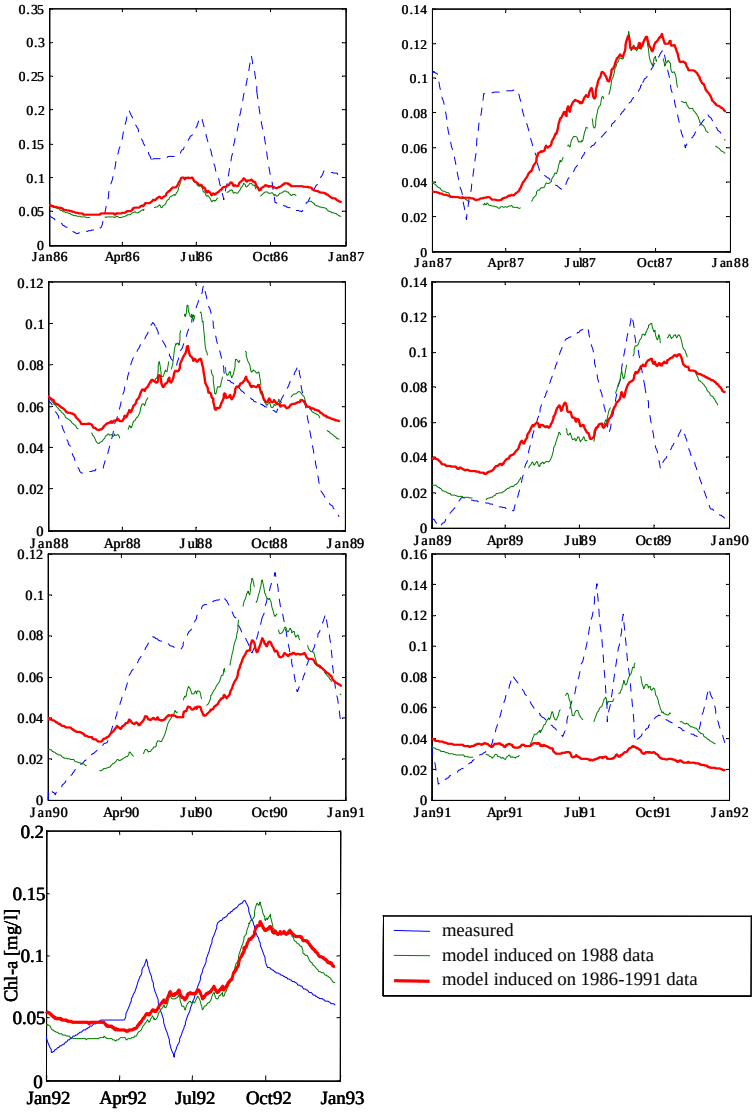
This equation (20.10) reflects that nutrient concentrations are supposed to have little impact on the growth process as expressed by the modified Monod kinetics in the first term. Half saturation constants in phosphorus and nitrogen limiting functions were calibrated by LAGRAMGE with very small values, i.e. 2.6E-07 and 9.8E-05, whereas silica has no influence at all with half saturation constant 0. The respiration process is formulated by simple first order kinetics. Mortality and sedimentation, i.e. the last two terms are formulated as temperature dependant processes. Equation (20.10) has a similar structure as equation (20.3) to (20.9) but different parameter values. However in contrast to the equation (20.5) discovered for the year 1988 it does not consider silica as limiting nutrient.

As the numerical solution of ODE requires initial values for each state variable we provided the measured initial values for the first day of each year in case that we simulated consecutive years as for experiment 2.

The Fig. 20.6 illustrates the simulation results of experiment 2 where the ODE structure and parameters for chl_*a* were assembled and adapted according to the Lake Kasumigaura data from 1986 to 1991, and tested for data of 1992. Though not very accurate the model still manages to predict most of the chl-a peaks and crashes. The simulation is best for years 1987 and 1988 and least accurate for 1986. The model quite accurately performs on the unseen data, i.e. data from 1992 (see Fig. 20.6).

In comparison with the model discovered on 1988 this model did not perform so well, though it was trained on longer data set. Possible explanation of this is that there is more noise in the long data set so it is more difficult to learn the lake's behaviour (with the present optimisation method). On the other hand learning the behaviour from one year's data is much easier but the year should be

representative enough so the model can be evaluated on longer period, which was the case in this study. In any case, long term data set is needed in order to draw



some relevant conclusions.

Fig. 20.6. Simulation results of the chl-*a* [mg/l] equation (20.10) assembled and trained by Lake Kasumigaura data of 1986 to 1992 and tested by the data of 1992

20.3.3. Experiment 3

The experiment 3 was carried out with Lake Kasumigaura data from 1986 to 1988 for training and data of 1989 for testing by adding the grazing process to the ODE for chl_*a* according to the task specification in Tab. 20.4. As a result the equation (20.11) had been discovered with the lowest value of MDL:

$$\frac{dchla}{dt} = chla \cdot 0.107 \cdot \frac{ortp}{ortp + 4.7E - 10} \cdot \frac{no3}{no3 + 0.00016} \cdot \frac{silica^2}{silica^2 + 0.01} \cdot \frac{temp}{9.3} \cdot \frac{light}{light + 147.6} - chla \cdot 0.054 \cdot \frac{temp - 2.4}{15 - 5} -$$
$$- chla \cdot \frac{0.009}{5} \cdot \frac{temp}{4.6} - clad \cdot 0.12 \cdot \frac{temp}{9.53} \cdot \frac{chla}{chla + 0} \cdot chla \cdot 0.07 \qquad (20.11)$$

In equation (20.11) the impact of nutrients on the growth process appears to be strengthened compared to equation (20.10). The grazing rate (Feeds_on) has been formulated by a proportional relationship with zooplankton (clad) and phytoplankton (chl_*a*) biomass as well as water temperature. The constant parameter value 0.07 indicates that only small amount of chl-a is consumed by zooplankton grazing. The training results of equation (20.11) achieve better MSE and MDL values in particular for 1986 (see Fig. 20.7) when compared with the previous equation (20.10). The equation (20.10) simulated well the seasonal dynamics of chl_*a* in 1986 and 1988 but overestimated the magnites of the late summer peaks. It didn't simulate well the chl_a dynamics in 1987 and 1989 by underestimating the spring and early summer peaks of both years and overestimating the late summer peak in 1987.

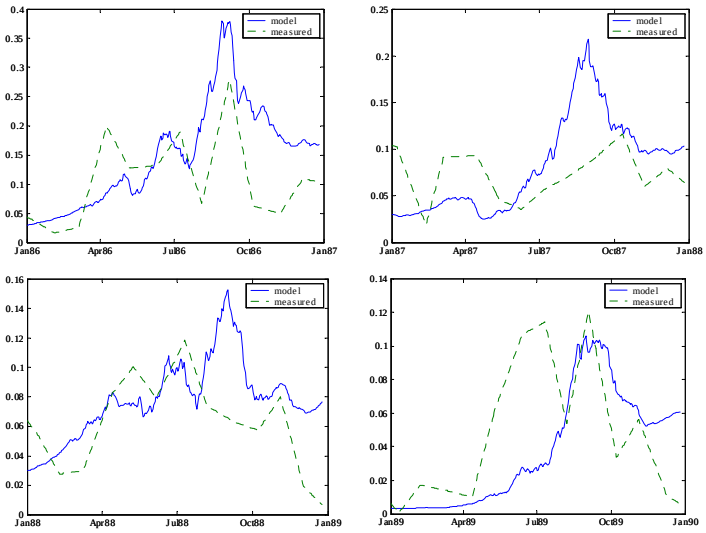


Fig. 20.7. Simulation results of the chl-*a* [mg/l] equation (20.11) annually assembled and trained by Lake Kasumigaura data of 1986 to 1988 and tested by the data of 1989

From experiment 3 it can be concluded that LAGRAMGE could not assemble a reasonable chl_*a* equation from data of one year only that would accurately simulate chl_*a* of other years. However better simulation results with lower MDL values were achieved by chl_*a* equations assembled and trained separately by data of each year i.e. equation (20.12) for 1986, (20.13) for 1987, (20.14) for 1988 and (20.15) for 1989 (see Fig. 20.8). These models performed achieved fairly good simulation results for the years 1988 and 1989, but underestimated spring peaks and overestimated autumn peaks in 1986 and 87. As expected the equations (20.12) to (20.15) show that rate functions for growth, respiration and sedimentation are differently represented when grazing is added to the chl_a mass balance equations. As a result the growth rates consider differently limiting nutrients, i.e. in 1986 phosphorus is identified in addition to nitrogen and silica, in 1987 all three nutrients remain limiting, in 1988 nitrogen is added to silica, and in 1989 all nutrients are limiting.

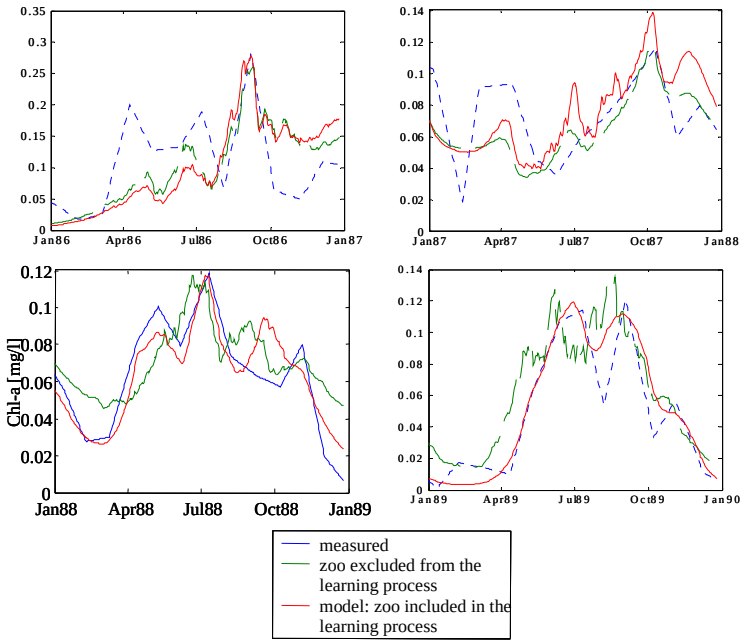


Fig. 20.8. Simulation results of the chl-*a* [mg/l] equations (20.12) to (20.15) annually assembled and trained by Lake Kasumigaura data of 1986 to 1989

This findings approve food web structures represented by deterministic lake models such as SALMO (Benndorf and Recknagel 1982; Recknagel and Benndorf 1982) that represent different functional algal groups such as diatoms, green and blue-green algae by separate ODE in order to realistically consider their specific nutrient requirements and selective preferencees during grazing by herbivorous zooplankton such as cladocera.

$$\frac{dchla}{dt} = chla \cdot 0.189 \cdot \frac{ps}{ortp + 4.7E-10} \cdot \frac{no3}{no3 + 1.84E-5} \cdot \frac{silica}{silica + 0.06} \cdot \frac{temp}{17.6 - 4.2} \cdot \frac{light}{light + 82.4} - chla \cdot 0.07 \cdot \frac{temp - 5}{16.6 - 3.9} -$$
$$- chla \cdot \frac{0.08}{5} \cdot \frac{temp}{5.1} - clad \cdot 0.2 \cdot \frac{temp}{5.7} \cdot \frac{chla}{chla + 0} \cdot chla \cdot 0.05$$

(20.12)

$$\frac{dchla}{dt} = chla \cdot 0.042 \cdot \frac{ortp^2}{ortp^2 + 6.2E-6} \cdot \frac{no3^2}{no3^2 + 1.9E-6} \cdot \frac{silica^2}{silica^2 + 0.016} \cdot \frac{temp}{5.8} \cdot light \cdot e^{\frac{\left(\frac{light}{100}+1\right)}{10.2}} - chla \cdot 0.01 \cdot 1.11^{(temp-17.9)} -$$
$$- chla \cdot 0.025 \frac{temp}{10.8} - chla \cdot \frac{0.04}{5} - clad \cdot 11.6 \cdot \frac{temp}{1} \cdot \frac{chla^2}{chla^2 + 0.9} \cdot chla \cdot 0.02$$

(20.13)

$$\frac{dchla}{dt} = chla \cdot 0.21 \cdot \frac{ortp}{ortp + 0} \cdot \frac{no3^2}{no3^2 + 4E-7} \cdot \frac{silica}{silica + 0.08} \cdot \frac{temp - 4.8}{20 - 0} \cdot \frac{light}{light + 15.3} - chla \cdot 0.038 \cdot 1.11^{(temp-19.5)} - chla \cdot 0.005 \cdot \frac{temp}{5.5} -$$
$$- chla \cdot \frac{0.095}{5} \cdot 1.11^{(temp-17)} - clad \cdot 0.47 \cdot \frac{temp - 0}{15 - 5} \cdot \frac{chla}{chla + 0} \cdot chla \cdot 0.23$$

(20.14)

$$\frac{dchla}{dt} = chla \cdot 0.17 \cdot \frac{ortp}{ortp + 1.2E-9} \cdot \frac{no3^2}{no3^2 + 2.1E-8} \cdot \frac{silica}{silica + 0.63} \cdot \frac{temp}{14.2} \cdot \frac{light}{light + 3.8} - chla \cdot 0.046 \cdot 1.13^{(temp-15)} -$$
$$chla \cdot \frac{0.32}{5} - clad \cdot 0.11 \cdot \frac{temp}{2.6} \cdot \frac{chla}{chla + 0.0005} \cdot chla \cdot 0.13$$

(20.15)

20.4. Conclusions

The software LAGRAMGE for computational assemblage and adaptation of ODE by using the expert knowledge and measured data has been applied for the simulation of chl-*a* in Lake Kasumigaura. As a result two types of chl-*a* models were discovered: (1) chl-*a* equations without considering zooplankton grazing assembled and trained by data of consecutive years were data of the last year was used for testing, and (2) chl-*a* equations considering zooplankton grazing assembled and trained by data of the years 1986 to 1989. The test results of the different models have demonstrated that LAGRAMGE can discover ODE that allow to simulate chl-*a* in Lake Kasumigaura for a variety of years. However the generalisation of discovered equations for unseen data of consecutive years was unsatisfactory, and the accuracy of calculated trajectories with regards to timing and magnitudes of peak events was moderate. The results have highlighted the importance of nutrients as growth limiting factors, and the need for considering functional algae groups in order to appropriately represent their selective grazing

by zooplankton.

References

Atanasova N (2004)

Benndorf J, Recknagel F (1982) Problems of application of the ecological model SALMO to lakes and reservoirs having various trophic states. Ecological Modelling 17, 129-145

Bobbin J, Recknagel F (2003) Evolving rules for the prediction and explanation of blue-green algal succession in lakes by evolutionary computation. In: Recknagel F (ed.) (2003) Ecological Informatics. Understanding Ecology by Biologically-Inspired Computation. Springer-Verlag Berlin, Heidelberg, New York, 291-310

Chapra SC (1997) Surface Water-Quality Modeling: McGraw-Hill

DeAngelis DL (1992) Dynamics of Nutrient Cycling and Food Webs. London: Chapmann & Hall

Dzeroski S, Todorovski L (2003) Learning population dynamics models from data and domain knowledge. Ecological Modelling, 170(2-3): 129-140.

Joergensen SE, Bendoricchio G (2001) Fundamentals of Ecological Modelling, Third Ed. Amsterdam: Elsevier Science Ltd.

Kompare B (1995) The Use of Artificial Intelligence in Ecological Modelling. Ph.D. Thesis, FGG, Ljubljana; Royal Danish School of Pharmacy, Copenhagen, Ljubljana, Copenhagen

Langley P, Sanchez J, Todorovski L, Dzeroski S (2002) Inducing process models from continuous data. Paper presented at the The Nineteenth International Conference on Machine Learning, Sydney Australia.

Recknagel F (1989) Applied Systems Ecology. Approach and Case Studies in Aquatic Ecology. Akademie-Verlag, Berlin, 1-138

Recknagel F, Bobbin J, Whigham P, Wilson H (2002) Comparative application of artificial neural networks and genetic algorithms for multivariate time-series modelling of algal blooms in freshwater lakes. Journal of Hydroinformatics 4, 2, 125-134

Recknagel F, Fukushima T, Hanazato T, Takamura N, Wilson H (1998) Modelling and prediction of phyto- and zooplankton dynamics in Lake Kasumigaura by artificial neural networks. Lakes & Reservoirs 3,123-133

Recknagel F (1997) ANNA - Artificial Neural Network model predicting species abundance and succession of blue-green Algae. Hydrobiologia, 349, 47-57

Recknagel F, French M, Harkonen P, Yabunaka K (1997) Artificial neural network approach for modelling and prediction of algal blooms. Ecological Modelling 96, 1-3, 11-28

Recknagel F,. Benndorf J (1982) Validation of the ecological simulation model SALMO. Int. Revue ges. Hydrobiol. 67, 1, 113-125

Straskraba M, Gnauck A (1985) Freshwater Ecosystems, Modelling and Simulation. Elsevier, Amsterdam

Todorovski L (2003) Using Domain Knowledge for Automated Modeling of Dynamic Systems with Equation Discovery. PhD Thesis, University of Ljubljana, Ljubljana, Slovenia

Todorovski L, Džeroski S (1997) Declarative bias in equation discovery. Paper presented at the 14th International Conference on Machine Learning, San Mateo, CA

Vollenweider RA (1968) The scientific basis of lake and stream eutrophication with

particular reference to phosphorus and nitrogen as eutrophication factors. Paris: Organisation for Economic Cooperation and Development.
Wei B, Sugiura N, Maekawa T (2001) Use of artificial neural networks in the prediction of algal blooms. Water Research, 35(8): 2022-2028
Whigham P, Recknagel F (2001) Predicting Chlorophyll-a in Freshwater Lakes by Hybridising Process-Based Models and Genetic Algorithms. Ecol. Modelling 146, 1-3, 243-251

Application of automated model discovery from data and expert knowledge to a real world domain: lake Glumsø

Nataša Atanasova[1], Ljupčo Todorovski[2], Sašo Džeroski[2], Boris Kompare[1]

[1] *Faculty of Civil and geodetic Engineering, University of Ljubljana, Slovenia*
[2] *Jožef Štefan Institute, Slovenia.*

Abstract
A novel approach to automated modelling (Lagramge) of lakes has been applied on lake Glumsø. The approach is based on the introduction of the expert knowledge in automated model induction from data. The method supports modelling with ordinary differential equations by following the mass conservation law. In both case studies Lagramge was used for discovering phytoplankton models. The data set for Lake Glumsø comprised two years daily measurements of data needed for food web modelling in lake. Using the expert knowledge a phytoplankton model was discovered from the data measured in the first year and evaluated on the second year data. The resulting model show good performance on the evaluation data set.

1. Introduction

Lake ecosystems are complex dynamic systems. Modelling of such ecosystems is a great challenge to the scientists, who are progressively improving and making more and more complex models. In general, we distinguish between two basic approaches to mathematical modelling. Following the deductive approach (knowledge driven), the model is derived from the basic background knowledge (e.g. basic physical, chemical and biological principles) from the domain of use. The second, inductive approach (data driven), is based on exploring some space of candidate models and face them against measured data. The model that fits measured data best is the result of the induction.

In this paper we apply an approach to modelling, which combines advantages of both, the domain expert knowledge and induction from measured data. The domain knowledge is gathered in a knowledge library, which is used to guide the process of induction. The result is a set of elementary models, mainly generic processes' descriptions, that follow the basic principles in the domain of interest (Todorovski and Dzeroski, 2001; Langley et Al., 2002; Todorovski, 2003). In the early days of the development of these tools (Todorovski & Džeroski, 1997), the knowledge had to be provided as an explicit definition of the space of candidate models. Now, these tools allow the user to provide higher-level domain knowledge about building mathematical models of complex real-world systems.

In this paper we focus on the application of the newly developed knowledge library

about water ecosystems on a real-world domain, i.e. lake Glumsø, Denmark. Lake Glumsø has been tackled with machine learning methods previously (Todorovski et. al. 1998, Todorovski 2003). Earlier version of Lagramge, i.e the version V 1.0 that required a hand crafted grammar has been used to discover a phytoplankton model. The same model was (re)discovered with the latest version V 2.0 of Lagramge (Todorovski 2003). However the model was discovered using a simple knowledge library. Slightly different model was discovered by implementing a complex knowledge library (Atanasova et Al., 2005). All of these experiments were performed on a small data set that did not allow for model evaluation. Just recently we obtained additional data for lake Glumsø (Jørgensen, 2004), i.e. a two year data set on which model evaluation was performed as well.

The paper is organized as follows: in the next chapter we briefly explain the method and the procedure of introduction of the expert knowledge about a specific ecosystem to the model discovery tool, i.e. Lagramge (if not specified, version V 2.0 is meant). In chapter three we present the data set and the experiments performed. Chapter four gives the results and discussion. Finally, the last chapter gives the conclusions.

2. The method: automated modelling framework

The procedure of automated modelling using the submitted, i.e. measured and suitably (re)interpreted data (see Kompare, 1995; Atanasova et al., 2005) on the one side and the background knowledge on the other side is shown in Figure 1. The modelling knowledge is gathered in a library of domain-specific knowledge. Next, modelling task has to be defined. This is done (in present version still manually) by user's specification of the observed system variables and processes that are expected to influence the behaviour of the system. Given a specification of modelling task at hand, Lagramge preprocessor can transform the high-level knowledge from the library into an operational form of a grammar. This grammar now completely specifies the space of candidate models of the observed system. This is illustrated in the left-hand side of Figure 1.

Once we have the grammar, we can use equation discovery system Lagramge to heuristically search through the space of candidate models, match each of them to submitted data by fitting the values of the constant parameters. These models evaluated (sorted) by two error measurements, i.e. mean square error (MSE) and MDL are the output of Lagramge. Further details about the modeling framework from Figure 1 can be found in (Todorovski, 2003).

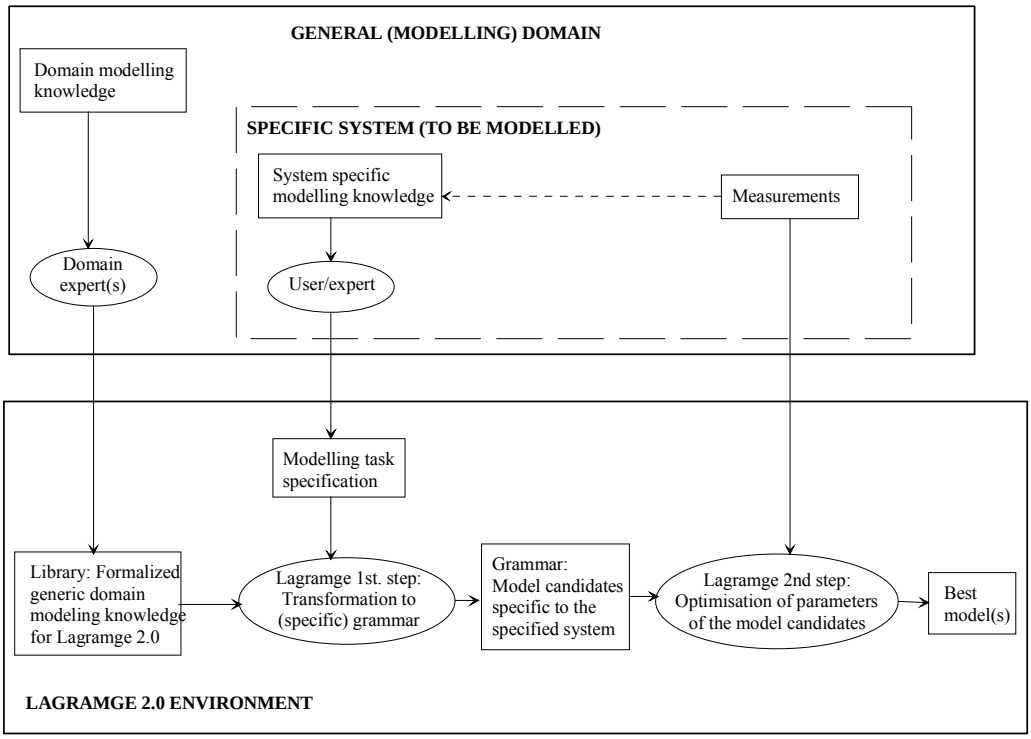


Figure 1: An automated modeling framework based on the integration of domain-specific modeling knowledge in the process of equation discovery

In order to be used in the model induction procedure, the knowledge needs to be coded in the knowledge library. Todorovski (2003) developed the formalism for encoding the domain knowledge about lakes' ecosystems. Using this formalism Atanasova et al. (2004) developed a comprehensive knowledge library for modelling food webs in lakes. The library supports construction of 0-dimensional N-box models, i.e., supports modelling of stratified lakes. It was estimated that the knowledge coded in the library covers great number of known lake models. Models of different complexity can be derived from the library, such as the simple Vollenweider's model (Vollenweider, 1968) or the fairly complex SALMO model (Bendorf, 1979 and Recknagel, 1980). For more details see Atanasova et al. (2004).

The knowledge library comprises general knowledge about modelling of lakes. In the task specification the expert (user of Lagramge) introduces the knowledge for a particulate observed ecosystem to the model discovery tool. The task specification includes declaration of the variables and processes in the system to be modelled. This will be explained more in detail in the following sections.

3. The data set and the experiments

Lake Glumsø (Jørgensen et al., 1986) is situated in a sub-glacial valley in Denmark. It is shallow with average depth of about 2 m. Its surface area is 266,000 m^2. For several years, it was receiving mechanically-biologically treated waste water from a community

with 3,000 inhabitants and a surrounding area which was mainly agricultural with almost no industry. The high nitrogen and phosphorus concentration in the treated waste water has caused hypereutrophication. The lake contained no submerged vegetation, probably due to the low transparency of the water and oxygen deficit at the bottom of the lake.

The new data set (provided by Jørgensen, 2004) includes two years of daily measurements from April, 1973 to April, 1974 and from October 1974 to October, 1975. For the experiments with the old data set see Atanasova et al., 2004. The new data set contains daily measurements of through flow, daily sunlight intensity [J/(cm^2*day)], water temperature, inorganic nutrients (dissolved phosphorus and nitrogen) in [mg/ℓ], phytoplankton expressed as Chl-a in [mg/ℓ] and zooplankton concentration in [mg DW/ℓ].

The experiment was set to discover a phytoplankton model (Chl-a), taking the soluble nutrients (ns and ps) and zooplankton (zoo) as data. The processes that affect phytoplankton concentration were considered to be growth of phytoplankton, respiration, settling and grazing by zooplankton. From previous experiments we learned that soluble phosphorus, or. ortophospahte is limiting growth, while soluble nitrogen is always abundant. Grazing represents a predatory loss of phytoplankton. It is influenced by temperature and by the phytoplankton and zooplankton concentrations. This specific knowledge about the processes was introduced to Lagramge through task specification as shown in Table 1.

Table 1: Task specification for the lake Glumsø

1:	variable Inorganic ns
2:	variable Inorganic ps
3:	system variable Primary_producer phyto
4:	variable Animal zoo
5:	variable Temperature temp
6:	process PP_growth(phyto, {ps}, {temp}, {light}) p1
7:	process Feeds_on(zoo, {phyto}, {temp}) p3
8:	process Respiration_PP(phyto, {temp}, {ps}, {light}) resp0
9:	process Sedimentation(phyto, {temp}) sed0

In the lines from 1 to 5 the variable types are declared, i.e. ns (dissolved inorganic nitrogen), ps (dissolved inorganic phosphorus) phyto (phytoplankton, expressed as Chl-a)), zoo (zooplankton) and temp (temperature). Processes are defined in the lines from 6 to 9. Phytoplankton growth is described in line 6. The process name is PP_growth and it has four arguments. The first is the name of the phytoplankton state variable. The arguments in the {} brackets, i.e. {ps}, {light} and {temp} define the influences and limitations of the process by nutrients, light and temperature respectively. Leaving one of them out would indicate no influence by the variable which was left out. For

instance, the definition of a growth process that is influenced only by the temperature and by two nutrients simultaneously (ortp and nitro) but not light limited, would be:

process PP_growth(phyto, {ps, ns}, {temp}, {})

The process Feeds_on (line 7) stands for (1) predatory loss of phytoplankton and (2) growth of zooplankton (zoo). Optional arguments of this process are the food (phyto) and temperature (temp), which means that the growth of zoo can be or not influenced by the food (none or many species) and temperature. Similarly, the rest of the processes in the system (respiration_PP, and Sedimentation) are defined (see lines 8 and 9).

Using the knowledge and the data set from October, 1974 to October, 1975 Lagramge was set to discover a phytoplankton model, using phosphorus and zooplankton as independent variables. The model was validated on the measurements from April, 1973 to April, 1974.

4. Results and discussion

Lagramge discovered a total phytoplankton model (eq 1) using the expert knowledge specified in Table 1 and the data set measured from October, 1974 to October, 1975.

$$\frac{dphyto}{dt} = phyto \cdot 0.18 \cdot \frac{ps^2}{ps^2 + 0.0012} \cdot \frac{temp}{15.4} \cdot light \cdot e^{-\frac{(\frac{light}{116.6}+1)}{116.6}} - phyto \cdot 0.01 - phyto \cdot \frac{0.36}{2} \cdot \frac{temp-4}{18-4}$$

$$-zoo \cdot 0.14 \cdot 1.13^{(temp-19)} \cdot \frac{phyto^2}{phyto^2 + 0.44} \cdot phyto \cdot 0.007$$

eq 1

The first term in the equation represents the phytoplankton growth term. The growth rate is limited with inorganic nutrient (phosphorus), temperature and light. Temperature influence is expressed with a linear function, while light limitation with the photoinhibition curve. Optimal light intensity is found to be 116.6 [J/(cm^2*day)]. Respiration is formulated with simple first order kinetics (second term in the equation). The third term is sedimentation, which is influenced by temperature with a linear temperature response curve. Finally, the last term in the equation represents a loss of phytoplankton due to grazing by zooplankton. The process is formulated using the filtration coefficient (0.14 [l/(mg zoo*day)]), exponential temperature curve and food (phytoplankton) limitation function. It is evident that only small portion of phytoplankton (0.007) is lost due to grazing. Model performance is shown on the left hand side of Figure 2. The data set measured from April, 1973 to April, 1974 was used for model validation on unseen data. This is shown on the right hand side of Figure 2.

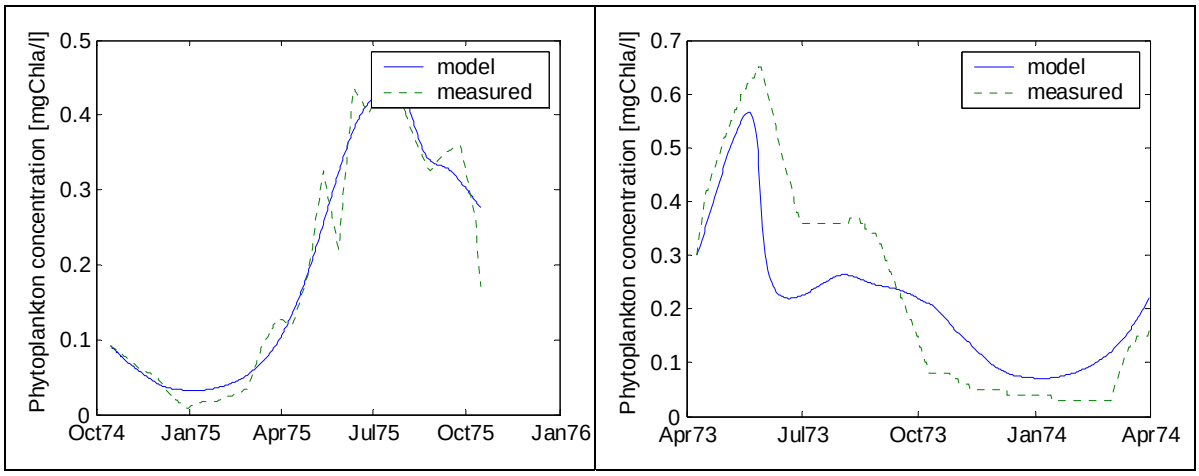


Figure 2: Phytoplankton model performance on lake Glumsø data. Left-side: performance on the training set and right-side: validation on unseen data

5. Conclusions

An approach to automated modelling, i.e. discovery of models in form of ODE's by using the expert knowledge and information from data, has been successfully applied on a real world domain lake Glumsø. The approach (Todorovski, 2003) is based on introduction of the expert knowledge about the system to be modelled in automated model induction from data. The data set comprised two years daily measurements of data needed for food web modelling in the lake. Using the expert knowledge a phytoplankton model was discovered from the data measured in the first year. The model evaluation on the second year data showed good fit to the measurements.

6. References

Atanasova, N., Todorovski, L., Džeroski, S., Kompare, B. 2005. Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *Ecological Modelling*. Accepted.

Kompare, B. 1995. *The Use of Artificial Intelligence in Ecological Modelling*. Ph.D. Thesis, FGG, Ljubljana; Royal Danish School of Pharmacy, Copenhagen, Ljubljana, Copenhagen.

Jørgensen, S. E. (2004): Lake Glumsoe data. In B. Kompare and N. Atanasova (Eds).

Langley, P., Sanchez, J., Todorovski, L. & Dzeroski, S. 2002. *Inducing process models from continuous data*. Paper presented at the The Nineteenth International Conference on Machine Learning, Sydney Australia.

Todorovski, L. & Dzeroski, S. 2001. *Using Domain Knowledge on Population Dynamics Modeling for Equation Discovery*. Paper presented at the Proceedings of the Twelth European Conference on Machine Learning, Freiburg, Germany.

Todorovski, L., Dzeroski, S. & Kompare, B. 1998. Modelling and prediction of phytoplankton growth with equation discovery. *Ecological Modelling*, 113(1-3): 71-81.

Todorovski, L. 2003. *Using Domain Knowledge for Automated Modeling of Dynamic Systems with Equation Discovery*. , University of Ljubljana, Ljubljana, Slovenia.

Todorovski, L. & Džeroski, S. 1997. *Declarative bias in equation discovery*. Paper presented at the 14th International Conference on Machine Learning, San Mateo, CA.

Automated modelling of a food web in lake Bled using measured data and a library of domain knowledge

Nataša Atanasova[1], Ljupčo Todorovski[2], Sašo Džeroski[2], Špela Rekar Remec[3], Friedrich Recknagel[4], and Boris Kompare[1]

[1]*Faculty of Civil and Geodetic Engineering, University of Ljubljana, Slovenia*
[2]*Jožef Stefan Institute, Slovenia*
[3]*Envoronmental Agency of the Republic of Slovenia*
[4]*University of Adelaide, Australia*

Abstract
In this paper, we applied automated modelling (computer model construction) method to the task of modelling a complex lake ecosystem. The method (Lagramge) integrates domain expert knowledge in the process of automated model induction from given data set. The data set comprises long term measurements (from 1995 to 2002) of physical, chemical and biological data in lake Bled, Slovenia. Given expert knowledge in terms of a simple food-web concept and rules for modelling thereof, we first induced a model for long-term dynamics of the phytoplankton in the lake. Failing to obtain a good fit, we also induced models of phytoplankton dynamics for each year separately. The differences between these models indicate structural dynamics of the food-web in lake Bled, i.e., indicate that the behaviour of the lake is changing from year to year. Additionally we successfully induced a three equation model (nutrient-phytoplankton-zooplankton) on the data from year 1996.

1. Introduction

Lake Bled has been a subject of exploration since late 1950-ies when first indices of eutrophication became obvious (Sketelj and Rejic, 1958). At early stages the research focused on measurements and observations, which among other showed very high dynamicity of the lake behaviour and rank it among complex ecosystems. Rismal (1980) set the first model of the lake using a stationary version of the Imboden's model (Imboden, 1974), which he improved to obtain inflow and outflow from each layer in order to simulate various proposed restoration measures, i.e., bringing additional fresh water into the hypolimnion, construction of a hypolimnetic siphon that takes the most nutrients' rich water from the bottom layers and reducing the nutrients' input to the lake. The benefits of the proposed siphoning were presented by a 2D and 3D hydrodynamic model (Rismal et al., 1997). Later Kompare (Kompare, 1995; Kompare et al., 1997) used machine learning techniques to model the lake's behaviour and to discover some additional knowledge from measured data. His models showed a typical three dimensional (3D) behaviour. Thus, the lake can not be modelled properly with 0D, 1-box models such as Vollenweider's model (Vollenweider, 1968), or 2-box model (Imboden, 1974; Rismal, 1980). According to this fact, physical segmentation of the system is required, i.e. at least 3-box (epi-, meta-, hypo-limnion) for each of the two basins (eastern and western). This research showed that we need a very complex

mathematical model to adequately describe this system.

On the other hand, regardless to their complexity, models represent not more than a simplified perception and understanding of the natural processes. Even if we know the concept of the system very well, we still have to solve a number of equations containing constants and parameters, which need to be estimated. Usually this leads to some numerical problems. Thus, we have to balance between too sophisticated models with many parameters, difficult to be estimated and too simple models with limited use.

Several questions emerge from this dilemma: (1) do we really need and can cope with such complex models, (2) is it possible to find a model structure that will properly cover the lake dynamics under all external conditions and for long period of time and (3) is the lake's system structure too dynamic for one model to cover the full long term behaviour? We offer some answers to these questions using an advanced machine learning technique Lagramge (Džeroski and Todorovski, 2003; Todorovski, 2003). Lagramge joins two fields in automated modelling, i.e. compositional modelling and a machine learning method, i.e. model induction from data. Compositional modelling builds models by assembling model fragments, typically from a library of model fragments, into an adequate model. In contrast, induction methods usually tackle the same task without incorporating domain expert knowledge in the procedure for model construction. The method used in the paper integrates the domain expert knowledge, gathered in a knowledge library (Atanasova et al., 2005), in the process of induction, performed by machine learning tools. This integration provides us with a guarantee that the constructed models will follow the basic principles from the domain of interest.

In the early days of the development of these methods (Todorovski and Džeroski, 1997), the knowledge had to be provided as an explicit specification of the space of candidate models. Now, Lagramge allow the user to provide higher-level (generic) knowledge about building mathematical models of complex real-world systems in the domain of interest (Todorovski, 2003). Given such library of knowledge and a specification of the modelling task, Lagramge first builds a specification of the space of candidate models and then, following the specification, searches for the model that follows the specification and fits measurement data best. Note that Lagramge searches for both optimal structure of the model as well as the optimal values of the model parameters.

2. The method: automated modelling framework

The machine learning method, used in this paper, supports introduction of the background modelling knowledge in the procedure of model induction from data. The knowledge provides recipe for building models in the domain of interest – it provides (1) taxonomy of basic process classes in the domain, (2) commonly used modelling alternatives for the processes in these classes, as well as (3) rules for combining the models of individual processes into the model of the whole observed system. Process classes represent a set of similar processes, for example, a process class “primary

producer growth" represents different types of growth processes including unlimited (exponential) growth, logistic (limited) growth, nutrient limited growth, etc. The knowledge library used here provides knowledge for modelling of food webs in lakes, following the mass conservation principle. The models are based on ordinary differential equations. For further details see (Atanasova et al., 2005).

In order to apply the modelling framework to a particular task of modelling a specific ecosystem, we have to provide modelling task specification, i.e., specification of the observed system variables and processes. Given a specification of modelling task at hand, Lagramge's pre-processor can transform the high-level knowledge from the library into an operational form of a grammar that specifies the space of candidate models of the observed system. Once we have the grammar, we can use equation discovery method Lagramge to heuristically search through the space of candidate models and match each of them to submitted data by fitting the values of the constant parameters. These models can be evaluated by two heuristic functions. One is mean square error (MSE) – it measures the discrepancy between measured data and data obtained by simulating the model. The other is minimum description length (MDL) function that takes into account model complexity and introduces preference towards simpler models.

As described above, the space of candidate models depends on knowledge library and modelling task specification. User can control the space of candidate models by providing different levels of detail about the model in the task specification. The detail level of the model definition can vary according to the expert knowledge about the observed system – the more structure is defined (or fixed) in the task specification, the smaller is the space of candidate models. This space is largest, if user only defines the state variables and does not specify any processes. In this case, Lagramge would search for models that are based on arbitrary combination of possible basic processes. If we know the relevant process classes for the observed system (or the particular system variable), we can further limit the space of candidate models to those that include these processes from those classes. Now, Lagramge will search for suitable process formulation within the specified process classes. Further limitation would include specification of the process formulation within the process class. In this case, the structure of the model is completely defined by the user and Lagramge performs only parameter calibration according to the given data set. Theoretically, we could even determine the parameters values and contract the search space to a single model, which would be a null task for Lagramge. This can be beneficial when we want to fix equations for only some of the system variables and let Lagramge to look for appropriate structure and parameters for the rest. Thus, the modelling formalism has the ability to complete a partially specified model.

Further details about the modelling framework can be found in (Todorovski, 2003; Atanasova et al., 2005).

3. Lake Bled data set

Lake Bled is a typical dimictic, subalpine lake of glacial-tectonic origin, situated in the NW Slovenia (14º 7' N and 46º 23'E), Europe. It occupies an area of 1.4 km^2 with a maximum depth of 30.1 m and an average depth of 17.9 m (Sketelj and Rejic, 1958). A sunken reef in the north-south direction at the position of the Bled island divides the lake into two basins - eastern and western (see Figure 1). The morphological characteristics of the lake are shown in Table 1. The monitoring of Lake Bled has been a part of the Slovene National Water-Quality Monitoring Programme since 1975. The data, obtained from the Ministry for Environment, Spatial Planning and Energy, Environmental Agency of The Republic of Slovenia, comprise long term (from 1987 to 2002) measurements of physical, chemical and biological parameters, but only the data from 1995 to 2002 are consistent and suitable for model induction. Samples are taken at two deepest locations in western and eastern basin, at every two meters from the surface to the bottom (Figure 1). During the periods when the lake surface was covered with ice, sampling was not performed. That is the reason why some data, especially data at the beginning or the end of the year, is missing. In 1995 and 1996 the sampling was performed all year around, since the lake was not ice covered, while in other years sampling usually started in March (or even April in 2001).

Table 1: Morphological characteristics of lake Bled

	Eastern basin	Western basin	Entire lake
volume [m^3 *10^6]	17.5	8.2	25.7
area [m^2 *10^6]	0.98	0.49	1.47
depth, max	24	30	30

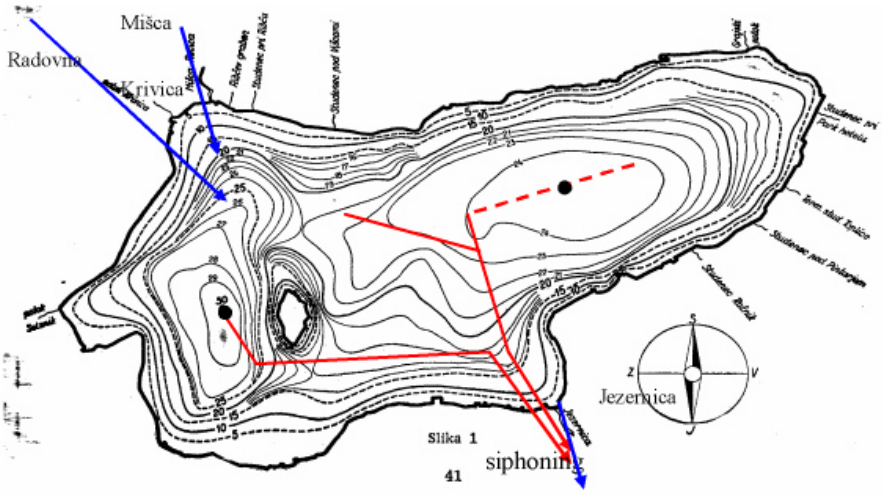


Figure 1: Lake Bled, adopted by (Sketelj and Rejic, 1958) and (Rismal, 1980). The dots on each side of the lake represent the sampling points

3.1 Physical data

The lake receives three major streams, i.e. the river Radovna, small torrent Krivica and the creek Mišca. There are also some minor inflows but for modelling purposes their influence was neglected. Flow rates of the inflows are measured daily, whereas the quality parameters of the streams are measured monthly. The lake has one natural outflow, the river Jezernica, and a siphoning outflow. The flow rates of both outflows are measured daily. Light is measured half-hourly, from 1993 as global radiation in W/m^2 at a location near the lake. Water temperature and water transparency are measured monthly.

3.2 Chemical and biological data about the lake

Samples for physical, chemical and biological analyses were taken in the period from 1995 to 2002 in the eastern and western lake basins monthly, at two metre intervals through the water column from the surface to 30 m at the western, and from the surface to 24 m at the eastern lake basins. Sampling from the depths was carried out by a Van Dorn bottle. The chemical analyses were carried out at the Environmental Agency of The Republic of Slovenia, following the standard methods.

The chemical data include measurements of inorganic nutrients important for algae modeling. These include concentrations of all forms of phosphorus (dissolved inorganic and total phosphorus), nitrogen (nitrate, nitrite and total nitrogen) and silica measured in [mass/volume].

The biological data include measurements of six taxonomical groups of phytoplankton and seven species of zooplankton. Their concentrations are measured in number of individuals per volume unit [No ind/ml]. In order to get compatible measurement units ([mass/volume]), we have to transform the measurement units to mg of dry weight (DW) per volume unit. While this transformation was already done for the total concentration of phytoplankton (Remec-Rekar, 1995), we used available information from literature and expert estimate to transform the measurement units of zooplankton. Of all the observed zooplankton species, only *Daphnia hyalina* (as most representative zooplankton species) was converted in [mass/volume] units. We estimated the average body length to be 2 mm and calculated the dry weight using the equation suggested by (Dumont et al., 1975). The list of all variables used for modelling is presented in Table 2. Note however, that for purposes of modelling phytoplankton change only (i.e., considering zooplankton to be an independent variable), this (approximate) transformation is not really necessary. To avoid it, we used zooplankton as measured (in [No ind/ml]), where possible.

3.3 Data preparation

Light, euphotic zone and temperature

Light was used as averaged daily value for underwater light in the euphotic (illuminated) zone. The depth of the euphotic zone was calculated from the measured

transparency in the lake:

$$z_{eu} = 1.7 \cdot transparency \quad (1)$$

The light extinction factor (k_e in [m^{-1}]) and the underwater light in the euphotic zone were calculated from the averaged daily global radiation (I), as follows:

$$k_e = \frac{4.6}{z_{eu}} \quad (2)$$

$$PAR = 0.5 \cdot I \quad (3)$$

$$PAR(z) = PAR \cdot e^{-k_e \cdot z} \quad (4)$$

$$light = avg(PAR(z)) \quad (5)$$

where *PAR* is photosynthetically available radiation, *z* is the water depth and *light* is depth averaged value for the underwater light in [J/cm^2*day] in the illuminated zone.

Daily water temperature data were obtained by a cubic spline interpolation over the monthly measured data.

Other data
As the majority of the measurements were performed on monthly basis we interpolated the daily data by cubic spline interpolation to get a convenient data set of "*daily*" measurements for induction of differential equations with Lagramge.

4. Experiments
The lake is naturally divided into an eastern and a western basin. According to the measurements the two basins have quite different characteristics and dynamics, which should be considered in the modelling procedure. Our modelling experiments refer to the eastern (bigger) basin and to the upper ten meters zone. No communication between the basins and between the upper and lower (hypolimnion) zone was taken into account.

Table 2: Measured data (variables) in lake Bled used for model induction

Variable name	Description	Units	Frequency
q_krivica	Inflow to the lake	m^3/day	Daily
q_misca	Inflow to the lake	m^3/day	Daily
q_radovna	Inflow to the lake	m^3/day	Daily
q_jezernica	Outflow (at surface)	m^3/day	Daily
q_natega	Outflow (syphon)	m^3/day	Daily
ps_krivica, ps_misca, ps_radovna	Nutrient (orthophosphate) concentration in the inflows	mg/l	Monthly
temp	Water temperature of the streams and lake	°C	Monthly
light	calculated underwater light	J/(cm^2 *day)	Monthly
ps, no, silica	Inorganic nutrients' concentration in the lake (ps is soluble phosphorus and no is nitrate)	mg/l	Monthly
phyto	Phytoplankton biomass concentration in the lake	mgDW/l	Monthly
daph	Zooplankton (*Daphnia hyalina*) biomass concentration in the lake	No ind/ml or mgDW/l (see text)	Monthly

4.1 Introducing the expert knowledge to Lagramge

We introduced modelling knowledge in the process of model discovery at two different levels. At the higher level is the general modelling knowledge about aquatic ecosystems (knowledge library) as described in (Atanasova et al., 2005). The lower level consists of a task specification that includes a list of variables and processes relevant for the modelling of lake Bled. The modelling task for lake Bled was introduced in a form of simple food-web concept shown in Figure 2. It includes three state variables, i.e. inorganic dissolved phosphorus, phytoplankton and zooplankton (daphnia hyalina) and the following processes: inflow/outflow of phosphorus, primary producer growth (PP_growth), predatory loss of phytoplankton (which is equal to the growth of daphnia (Feeds_on)), non-predatory loss of phytoplankton (Respiration_PP, Settling), nonpredatory loss of daphnia (Respiration_A) and mortality of daphnia (Mortality_A), which also accounts for daphnia predatory loss.

The knowledge library includes several formulations for each of above listed process classes (Atanasova et al., 2005). For example the process class PP_growth contains five different models for primary producer growth, i.e. exponential, logistic, growth limited by temperature, light, and nutrients, growth limited model that accounts for variable optimal temperature, as well as growth limited model that couples the effects of light and temperature. Furthermore light, temperature, and nutrients limitations are defined as

function classes that include several different formulations for each. Thus, we have more than fifty possible formulations for the PP_growth process. Similarly, we have a number of candidate formulations for the rest of the process classes in this lake.

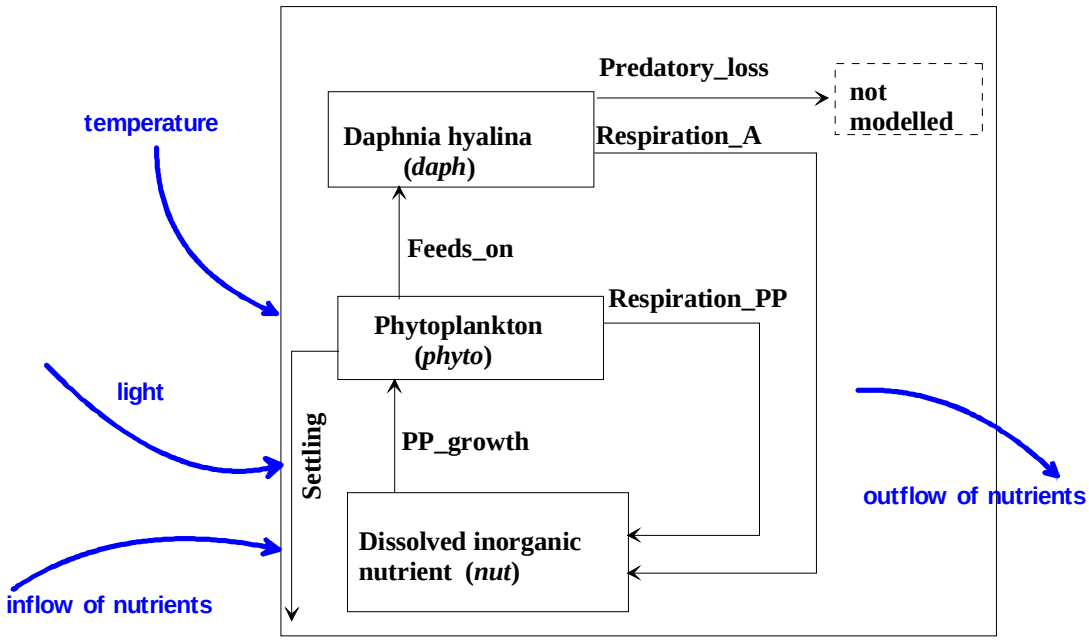


Figure 2: Simple conceptual model for lake Bled

The library of modelling knowledge also specifies how to combine the processes into a corresponding model of the whole system (Džeroski and Todorovski, 2003; Todorovski, 2003; Atanasova et al., 2005). The combining rules in the library support modelling with ordinary differential equations by following the mass conservation principle. More details about this kind of modelling can be found in (e.g. (Jørgensen and Bendoricchio, 2001) or (Chapra, 1997)). According to the combining rules from the library, the processes defined in the task specification, will be composed in the following model based on three differential equations (6, 7, and 8):

$$\frac{dnut}{dt} = \text{Inflow - Outflow} + const \cdot \text{Respiration\_PP} + const \cdot \text{Respiration\_A} - const \cdot \text{PP\_growth} \quad (6)$$

$$\frac{dphyto}{dt} = \text{PP\_growth - Respiration\_PP - Sedimentation - Feeds\_on} \quad (7)$$

$$\frac{ddaph}{dt} = const \cdot \text{Feeds\_on - Respiration\_A - Mortality\_A} \quad (8)$$

4.2 Discovering models

First, we made an attempt to discover a model that would describe the long term behaviour of the lake. For this, we used the task specification, described in the previous section with the only difference being that *phyto* is the only system variable, while daphnia and phosphorus were considered to be exogenous variables (i.e., forcing functions). Lagramge was then used to discover a specific model following equation 7 from the data for years 1995 to 2001.

Failing to get a very good fit to the long term data, we conjectured that the lake dynamics changes from year to year. In our second experiment, we aimed at testing this hypothesis, so we applied Lagramge to build separate models for each year data.

In the final experiment, we aimed at discovering a model that includes three system variables (phosphorus, phytoplankton, and zooplankton) from one year's data (1996). Due to the complexity of space of candidate models and limited computational resources[1] we decided to induce equation for each of the system variables at a time, following the food web hierarchy (phosphorus – phytoplankton – zooplankton). According to the mass balance for inorganic nutrient (see equation 6) following processes defined in the expert task definition were included: inflow of inorganic phosphorus, outflow, release of nutrients due to phytoplankton and zooplankton respiration, and loss due to phytoplankton growth.[2] The two processes that influence both the phosphorus and phytoplankton (equation 7) equation are PP_growth and Respiration_pp. Since the discovered equation for phosphorus already fixed the formulation for these processes, we used the same fixed formulation for discovering the phytoplankton equation and only search for appropriate formulation of the other processes involved there (i.e., sedimentation and predatory loss, Feeds_on). Similarly, when the phytoplankton equation is discovered, we used the already discovered formulation of the processes Respiration_A in the phosphorus equation and Feeds_on, in the phytoplankton equation, and let Lagramge find an appropriate formulation for the mortality of daphnia. Note finally, that the models induced in the last experiment involve zooplankton measured in [mgDW/l]. Experimental setup is summarized in Table 3.

[1] Note that induction of a single equation with Lagramge takes tens of hours of CPU time on the equipment (Pentium based Linux Platform with 2GHz processor and 1GB of RAM) we used for the experiments.

[2] We should point here that the recycling of nutrients goes through more stages (e.g. decomposition of dead organic matter, detritus), which were skipped here in favour of model simplicity.

Table 3: The experimental setup in lake Bled

	Experiment 1	Experiment 2	Experiment 3
system (target) variables	phyto	phyto	ps, phyto, daph [3]
independent variables (forcing functions)	temp, light, ps, no, silica, daph	temp, light, ps, no, silica, daph	q_radovna, q_krivica, q_misca, q_jezernica, q_natega, ps_radovna, ps_krivica, ps_misca, temp, light
training data set(s)	1995 to 2001	1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002	1996

5. Results

5.1. Long term phytoplankton model

Using the task specification from Section 3 and the data series from 1995 to 2001, Lagramge discovered the following model for phytoplankton (equation 9):

$$\frac{dphyto}{dt} = phyto \cdot 0.145 \cdot \frac{ps}{ps + 0.0006} \cdot \frac{silica}{silica + 0} \cdot \frac{no}{no + 0.01} \cdot 1.2^{temp-20} \cdot \frac{light}{200} \cdot e^{\left(1 - \frac{light}{132}\right)} -$$
$$- phyto^2 \cdot 0.002 \cdot \frac{temp - 0.5}{20} - phyto \cdot \frac{0.28}{10} - daph \cdot 0.96 \cdot \frac{temp}{6.5} \cdot \frac{phyto}{phyto + 19} \cdot 0 \qquad 9$$

The first term in equation 9 represents the growth process of phytoplankton, which is formulated as nutrient, temperature, and light limited. Nutrient limitation is modelled with the Monod term, where phosphorus and nitrate are found as limiting nutrients. Temperature influence on growth is modelled using the exponential adjustment curve, while light limitation on growth is modelled with the photoinhibition curve (Steele, 1965). Respiration of phytoplankton (the second term) is modelled with second order kinetics, where temperature influence is formulated with the linear response curve. Finally, the last two additive terms in the phytoplankton equation represent the settling process and the process of grazing by zooplankton (*daph*). According to the model the grazing term equals zero, i.e., grazing has no influence on the phytoplankton dynamics.

The comparison of measured and simulated phytoplankton concentration shows a poor fit (Figure 3) to the training as well as to the testing data set (data from year 2002). There are several possible reasons for this. First, we might need more complex model structure including several alga species and perhaps also the diet preferences of zooplankton. However, due to the limitations of measured data, this hypothesis can not be properly tested. Second reason might be that the lake dynamics changes through the time.

[3] the model of three differential equations was not discovered simultaneously (see text)

We can easily test this hypothesis by inducing models from data on individual lake cycles, i.e., calendar years. The test of this hypothesis is the objective of the second experiment.

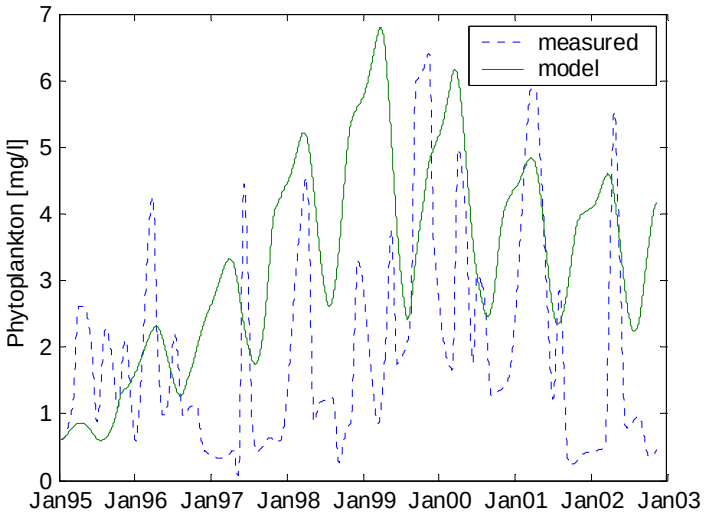


Figure 3: Long term simulation of the phytoplankton model (equation **9**)

5.2. Discovering a phytoplankton model for different years

In this experiment Lagramge was set to discover phytoplankton models with same basic structure (see equation 7), but for each year data from 1995 to 2002 separately.

Note that the induced models have different formulations of the processes and different parameter values. The growth term in all models is formulated as nutrient, temperature and light influenced process. Nutrient limitation functions for *ps*, *no3* and *silica* is formulated either with the two variations of the Monod term (i.e., $x / (x + const)$ and $x^2 / (x^2 + const)$) or with the exponential limiting term (i.e., $1 - e^{-const*x}$). Note that the smaller values of the constant (also called saturation coefficient) in the Monod terms indicate smaller influence by (nutrient) *x* on phytoplankton growth. In the limit, a term with saturation coefficient zero, (i.e., $x / (x + 0)$), the influence equals one, which means that phytoplankton growth is not limited by *x*. In contrast with Monod terms, exponential term behaviour is the opposite – larger constant parameter values correspond to smaller influence by *x* is (the term's value is closer to 1). Following these simple rules, we can interpret the discovered models in terms of the nutrients' influence on the total phytoplankton growth and analyze how this influence changes from year to year. Table 4 summarizes the types of influences in the induced equations (15 to 22 in the Appendix).

Analysis of the nutrient, light, and temperature influence on phytoplankton growth shows the following. In 1995, *silica* was found as the only limiting nutrient, in 1996 and

2001, *ps and silica*, in 2002 *silica and nitrate*, while in the period from 1997 to 2000 all of the nutrients were important (limiting) for the phytoplankton growth. It is interesting that in 1997 and 1998 light was found not to influence the phytoplankton growth. In the models for 1995 and 2002, light influence on growth was modelled with the Monod term (saturation curve), while in 1999 to 2001 by the photoinhibition curve (Steele, 1965). Temperature influence on growth is modelled either using the linear model (1997, 1998, and 2002) or the exponential one (1995, 1996, and 1999 to 2001).

Next, we analyzed the influence of respiration on the phytoplankton dynamics. It is modelled with first (1995, 1996, 1998, and 1999) or with second (in 1997 and 2000 to 2002) order kinetics. The respiration is temperature influenced in all years except for 1996.

Finally, the last two additive terms in the phytoplankton equation represent the settling process and the process of grazing by zooplankton (*daph*). According to the models in 1995, 1996, 1997, and 2001 the grazing term equals zero, i.e., grazing has no influence on the phytoplankton dynamics.

Table 4: Description of the variables influences on the phytoplankton dynamics equations induced on one year data sets (1995-2002). The influence is described using the following labels: no denotes no influence, yes denotes presence of influence, and other labels (exp, lin, mon, mon2) denote the specific influence model (exponential, linear, Monod, second order Monod, respectively).

process/term	growth					respiration	settling	
variable	temp	light	ps	silica	no	temp	temp	grazing
1995	exp	mon	no	mon	no	lin	lin	no
1996	exp	inh	mon	no	mon	no	no	no
1997	lin	no	mon	mon	mon	exp	no	no
1998	lin	no	exp	mon	exp	exp	no	yes
1999	exp	inh	mon	mon2	mon	lin	exp	yes
2000	exp	inh	mon	exp	mon	lin	no	yes
2001	exp	inh	mon2	exp	no	lin	lin	no
2002	lin	mon	no	mon2	mon2	lin	no	yes

The simulation of the models is compared with the measurement data in Figure 4. Note the different starting time of the simulations due to missing data in winter periods (see section 3). The models perform much better than the one induced on the data from the full time span (see Figure 3). The goodness of fit is evaluated by the root mean square error (RMSE). The best fit (lowest RMSE) is obtained on the 2002 data, and the worst one on the 1996. A possible reason for poor fit is that we took the nutrient and zooplankton data for granted, instead of treating these two variables as system variables. So, in the last experiment, we aimed at building a complete model of food web in the lake from the 1996 measurements.

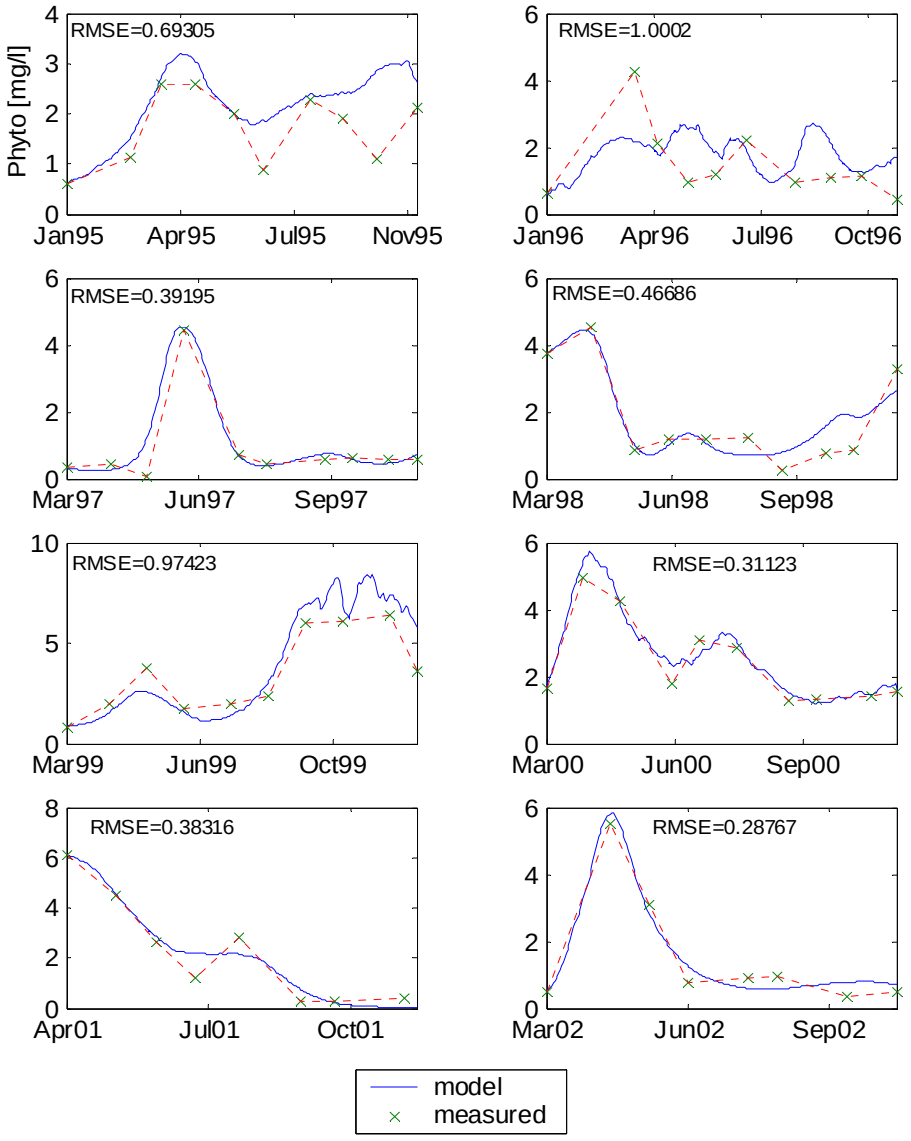


Figure 4: Performance of the phytoplankton models, discovered separately for each year data. The goodness of fit of each model is evaluated by the root mean square error (RMSE).

In addition, we validated each of the models discovered on specific year on unseen data measured in other years. The validation of almost all models revealed that there is a big discrepancy between the simulated and measured data, which indicates that we deal with a very complex system without yearly repeating patterns. Yet, the model induced on 2002 data shows fairly good performance on the other years (Figure 5). The model correctly follows the trend of phytoplankton dynamics in all years, except for year 1999. Note also that the model systematically overestimates the spring phytoplankton

peaks.

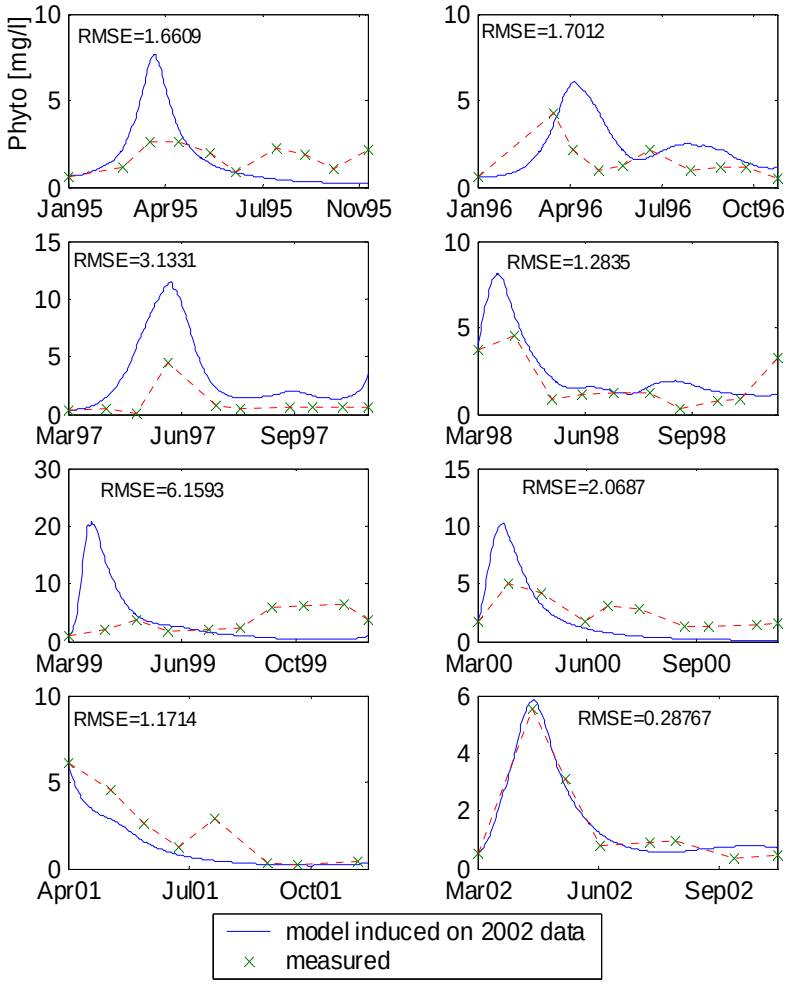


Figure 5: Validation of the phytoplankton model induced from 2002 data on the rest of the simulation period

5.3 Inducing basic food web model

As already explained in Chapter 4 this model was discovered gradually, one equation at a time, starting with phosphorus equation, continuing with the discovery of phytoplankton, and daphnia equation.

Phosphorus equation

When discovering this model phytoplankton and daphnia are taken as forcing functions (data). Using the task specification from Section 3 and the 1996 daily data Lagramge discovered several good phosphorus models having the form shown in equation 6. The one with the lowest error is shown below (10).

$$\frac{dps}{dt} = ps\_krivica \cdot \frac{q\_krivica}{7 \cdot 10^6} + ps\_misca \cdot \frac{q\_misca}{7 \cdot 10^6} + ps\_radovna \cdot \frac{q\_radovna}{7 \cdot 10^6}$$
$$- ps \cdot \frac{q\_jezernica}{7 \cdot 10^6} - ps \cdot \frac{q\_natega}{7 \cdot 10^6} + 0.0022 \cdot phyto^2 \cdot 0.072 \cdot \frac{temp - 2.7}{20.4 - 2.7} + 0.07 \cdot daph \cdot 0.0026 \cdot \frac{temp}{12.3} -$$
$$0.0023 \cdot phyto \cdot 0.21 \cdot \frac{ps}{ps + 0.00042} \cdot \frac{temp}{16.7} \cdot \frac{light}{170} \cdot e^{(1 - \frac{light}{170})}$$

10

The first five terms in the equation represent the inflow and outflow of inorganic phosphorus. Next term, phytoplankton respiration (i.e. release of phosphorus due to phytoplankton respiration), is formulated as second order kinetics, while daphnia respiration as first order kinetics. Both processes are temperature influenced. Growth of phytoplankton, i.e., consumption of phosphorus by phytoplankton is modelled as temperature, light, and nutrient limited growth. Nutrient limitation is modelled with Monod expression, light limitation with the photoinhibition curve (Steele, 1965) and temperature influence with linear response curve.

Phytoplankton equation

In the phytoplankton mass balance equation (7) following processes (equations 11 and 12) are already discovered in the phosphorus equation:

$$\text{PP\_growth} = phyto \cdot 0.21 \cdot \frac{ps}{ps + 0.00042} \cdot \frac{temp}{16.7} \cdot \frac{light}{170} \cdot e^{(1 - \frac{light}{170})}$$

11

$$\text{Respiration\_PP} = phyto^2 \cdot 0.072 \cdot \frac{temp - 2.7}{19.7 - 2}$$

12

According to this we set Lagramge to discover the rest of the processes in the phytoplankton equation, i.e., Sedimentation and Feeds_on (or grazing by daphnia). The best phytoplankton model using the growth and respiration terms from the phosphorus model is shown below (13).

$$\frac{dphyto}{dt} = phyto \cdot 0.21 \cdot \frac{ps}{ps + 0.00042} \cdot \frac{temp}{16.7} \cdot \frac{light}{170} \cdot e^{(1-\frac{light}{170})} - phyto^2 \cdot 0.072 \frac{temp - 2.7}{19.7 - 2} -$$
$$- phyto \cdot \frac{0.5}{10} \cdot \frac{temp - 2}{18 - 4} - daph \cdot 0.5 \cdot \frac{temp - 2.6}{18 - 4} \cdot (1 - \exp(-0.58 \cdot phyto)) \cdot 0.56 \cdot phyto$$

13

Sedimentation is formulated as temperature influenced, with sedimentation rate 0.5 m/day. The grazing term is formulated using the filtration coefficient (0.5 l/(mg*day)), linear temperature response curve and exponential term for food limitation on daphnia growth.

Comparison of the phytoplankton equation with the one from the previous Section (equation 16 in the appendix) shows the differences in practically all process' formulations. The phytoplankton growth is limited by phosphorus concentration and temperature, while the growth in equation (16) is limited by two nutrients (phosphorus and silica). Note also the difference in the temperature influence terms. Also, respiration is formulated with second order kinetics, (first order in equation 16) and sedimentation (third term) is temperature influenced (temperature independent in equation 16). Finally, the most important difference between two models is that grazing influence is important for the phytoplankton dynamics, unlike the previous experiment where the grazing term equals zero.

Considering that this model has better performance (i.e., lower RMSE, see Figure 6 and Figure 4), it is a bit of surprise, that Lagramge could not find a suitable set of parameters in the previous experiment. Obviously, the change of *daphnia* units pushed the parameters' values in a range where the optimization method is unsuccessful. Note that in this experiment *daphnia* is expressed in [mgDW/l], while in the previous in [No ind/ml].

Zooplankton (daphnia) equation
Zooplankton equation contains the following processes: Feeds_on, Respiration_A and Mortality_A. Feeds_on correspond to grazing of phytoplankton and therefore was already discovered in the phytoplankton equation, while Respiration_A is already discovered in the phosphorus equation. The task here is to find suitable mortality process for daphnia, which is a closure term for the model. The equation with lowest error is presented in (14). Lagramge found the hyperbolic term as the most suitable closure term for the model.

$$\frac{ddaph}{dt} = 0.14 \cdot daph \cdot 0.5 \cdot \frac{temp - 2.6}{18 - 4} \cdot (1 - \exp(-0.58 \cdot phyto)) \cdot 0.56 \cdot phyto$$
$$-daph \cdot 0.026 \cdot \frac{temp}{12.3} - 0.01 \cdot \frac{daph^2}{0.001 + daph}$$

14

Thus, the complete basic food web model for phosphorus, phytoplankton and daphnia consists of equations (10, 13, and 14). Simulation of the model is shown in Figure 6.

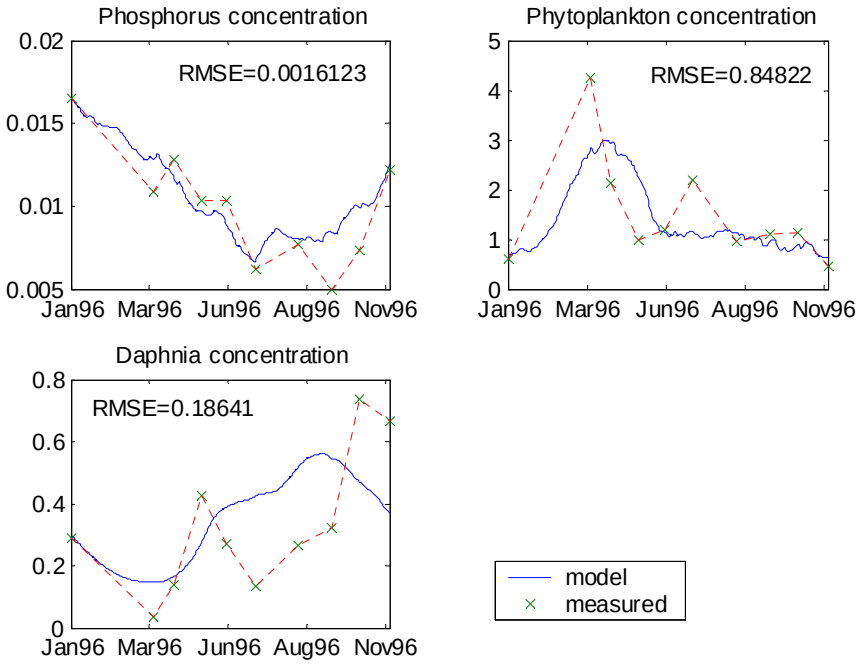


Figure 6: Performance of the food web model for phosphorus, phytoplankton, and daphnia

Though the model was successfully calibrated on 1996 data it shows problems when validating it on the other years' data sets. This behaviour yet again confirms our hypothesis that the lake dynamics changes yearly.

6. Conclusions and further work

Discovering a model based on ordinary differential equations that covers long term behaviour of such a complex system is very difficult task, mostly due to the system complexity and constantly changing patterns of the real system behaviour through time. Another issue is the complexity of the computational method itself, which is strongly limited by the present computational resources used in this research. Therefore, we limited our task on either discovering equation for one state variable instead of discovering the complete model simultaneously, or discovering a complete model with

strong limitations on the search space of candidate models.

The aim of discovering equation for one state variable instead of complete model of the system is above all to search connections and patterns among data. In our case the relationships were prescribed by domain expert – therefore they are transparent and understandable to experts. According to our expectations, we could not find a single model that will be suitable for such complex system over the whole time period. Comparison of the measurement data from different years shows that lake Bled usually has two peaks of algae bloom each year (spring and autumn) caused by different algae species. However, the situation in certain years can also be different: in 1995 we can notice three algal blooms, while in 1997 and 2002, we observe only one. On the other hand, modelling one year behaviour is a reasonable task, as it is evident from Figure 4.

For 1996, Lagramge successfully discovered a complete three equation food web model, though the search space of candidate models was strictly limited and controlled. The chosen induction order that follows the food web hierarchy is only one possible and plausible order used here as a heuristic. However, further experiments are required to evaluate the influence of induction ordering on the obtained model.

Phosphorus and phytoplankton equations were discovered very successfully as evident from the simulations on Figure 6. Discovering daphnia equation was more difficult task due to following reasons: (1) the conversion of data from number of individuals to biomass was approximated with literature data and (2) daphnia predation by fish was not modelled. Daphnia mortality was the closure term in the model. Of the four animal loss terms defined in the library, i.e. first order kinetics, second order kinetics, hyperbolic and sigmoid form, Lagramge found the hyperbolic form as the most suitable one. Simulation of the complete model indicates good fit with the measurements for phosphorus and phytoplankton, while for daphnia, the model only captures the trend and not the daphnia dynamics.

In order to find a complete model, that will cover the long term behaviour of the lake, several investigations still need to be done, to confirm some assumptions that emerged during this research. The first assumption is that the lake has dynamic structure and therefore we can not model it with a single model with constant parameters. This assumption was partly confirmed in the second experiment when Lagramge successfully discovered phytoplankton models, which were trained for each year separately (equations 15 to 22 in the appendix). The structure of all models was defined by expert (in the task specification) to have four processes, i.e. growth, respiration, sedimentation and grazing. To our expectations discovered models differ in the processes' formulations and in their parameters values which indicates the structural dynamicity of the lake, i.e. the system's structure is different from year to year.

Our second assumption is that the concept that we used for discovering models is too simple to cover long term behaviour of the lake. It is necessary to increase the

complexity of the concept by introducing the algal functional groups as state variables. For this we need more detailed insight into the lake food web and some more expert knowledge. In order to accomplish such demanding task we also need faster computational resources.

Finally, our last assumption is that improvement of the optimisation method for simultaneous multiple parameter estimation would result in better models even by using the current simple concept.

Acknowledgements
The data for this research were provided by the Ministry for Environment, Spatial Planning and Energy, Environmental Agency of The Republic of Slovenia. First author would like to acknowledge the Ministry of Education, Science and Sport for two year junior researcher grant No. 3311-02-831/433. We also acknowledge the support of the ECOGRID project, funded by the Slovenian Research Agency, under contract No. 3311-04-828125. Finally, thanks to the reviewers for their constructive comments on an earlier version of the manuscript.

References:

Atanasova, N., Todorovski, L., Džeroski, S. and Kompare, B. (2005): Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *Ecological Modelling*. Accepted.

Chapra, S. C. (1997): *Surface Water-Quality Modeling*. McGraw-Hill 0-07-011364-5.

Dumont, H. J., Van de Velde, I. and Dumont, S. (1975): The Dry Weight Estimate of Biomass in a Selection of Cladocera, Copepoda and Rotifera from the Plankton, Periphyton and Benthos of Continental Waters. *Oecologia (Berl.)* **19**, 75-97.

Džeroski, S. and Todorovski, L. (2003): Learning population dynamics models from data and domain knowledge. *Ecological Modelling* **170**, 2-3, 129-140.

Imboden, D. (1974): Phosphorus model of lake eutrophication. *Limnology and Oceanography* **19**, 297-304.

Jørgensen, S. E. and Bendoricchio, G. (2001): *Fundamentals of Ecological Modelling*. Elsevier 0-080-44028-2.

Kompare, B. (1995): The Use of Artificial Intelligence in Ecological Modelling: *Ljubljana, FGG; Royal Danish School of Pharmacy*, FGG, Ljubljana; Royal Danish School of Pharmacy, Copenhagen, Ljubljana, Copenhagen.

Kompare, B., Džeroski, S. and Karalic, A. (1997): Identification of the Lake Bled ecosystem with the artificial intelligence tools M5 and FORS. Fourth International Conference on Water Pollution. , pp. 798.

Remec-Rekar, S. (1995): Življenska strategija in absorbcija fosforja pri nekaterih fitoplanktonskih vrstah Blejskega jezera-123, University of Ljubljana, Ljubljana.

Rismal, M. (1980): Presoja posameznih metod za sanacijo Blejskega jezera. *Gradbeni vestnik* **29**, 2-3, 34-46.

Rismal, M., Kompare, B. and Rajar, R. (1997): Contribution of hydrodynamic and limnological modelling to the sanitation of Lake Bled. Fourth International Conference on Water Pollution. , pp. 139.

Sketelj, J. and Rejic, M. (1958): Preliminary account on the examination of Lake Bled. *Gradbeni vestnik* **61-64**.

Steele, J. (1965): Notes on Some Theoretical Problems in Production Ecology, pp. 393-398. In C. Goldman (Ed.): *Primary Production in Aquatic Environments.*, University of California Press, Berkeley, California.

Todorovski, L. (2003): Using Domain Knowledge for Automated Modeling of Dynamic Systems with Equation Discovery: *Fakulteta zaraèunalništvo in informatiko*, University of Ljubljana, Ljubljana, Slovenia.

Todorovski, L. and Džeroski, S. (1997): Declarative bias in equation discovery. 14th International Conference on Machine Learning. , pp. 384.

Vollenweider, R. A. (1968): The scientific basis of lake and stream eutrophication with particular reference to phosphorus and nitrogen as eutrophication factors, Organisation for Economic Cooperation and Development, Paris.

Appendix

1995:

$$\frac{dphyto}{dt} = phyto \cdot 0.25 \cdot \frac{ps}{ps+0} \cdot \frac{silica}{silica+0.34} \cdot \frac{no}{no+0} \cdot 1.12^{(temp-20)} \cdot \frac{light}{light+45.2} - phyto \cdot 0.07 \cdot \frac{temp-4.2}{15.5-4.4} -$$

$$- phyto \cdot \frac{0.0002}{10} \cdot \frac{temp-0}{15-5} - daph \cdot 10^{-5} \cdot \frac{temp}{18} \cdot \frac{phyto^2}{phyto^2+19.8} \cdot phyto \cdot 0$$

15

1996:

$$\frac{dphyto}{dt} = phyto \cdot 5 \cdot \frac{ps}{ps+0.0024} \cdot \frac{silica}{silica+0.19} \cdot \frac{no}{no+0} \cdot 1.11^{(temp-15)} \cdot \frac{light}{100} \cdot e^{\left(1-\frac{light}{100}\right)} - phyto \cdot 0.032 -$$

$$- phyto \cdot \frac{0.31}{10} - daph \cdot 10^{-5} \cdot \frac{temp}{7.8} \cdot \frac{phyto^2}{phyto^2+4.8} \cdot phyto \cdot 0$$

16

1997:

$$\frac{dphyto}{dt} = phyto \cdot 5.04 \cdot \frac{ps}{ps+0.0005} \cdot \frac{silica^2}{silica^2+0.688} \cdot \frac{no^2}{no^2+15} \cdot \frac{temp-5}{16.4-4.3} \cdot \frac{light}{light+0} - phyto^2 \cdot 0.058 \cdot 1.13^{(temp-16)} -$$

$$- phyto \cdot \frac{0.43}{10} - daph \cdot 0.0063 \cdot \frac{temp}{18} \cdot \frac{phyto^2}{phyto^2+11.7} \cdot phyto \cdot 0$$

17

1998:

$$\frac{dphyto}{dt} = phyto \cdot 4.66 \cdot (1-e^{-4.5 \cdot ps}) \cdot \frac{silica}{silica+0.15} \cdot (1-e^{-3.3 \cdot no}) \cdot \frac{temp}{10.2} \cdot \frac{light}{light+0} - phyto \cdot 0.32 \cdot 1.12^{(temp-17.7)} -$$

$$- phyto \cdot \frac{0.5}{10} - daph \cdot 15 \cdot \frac{temp}{15-5} \cdot \frac{phyto^2}{phyto^2+20} \cdot phyto \cdot 0.79$$

18

1999:

$$\frac{dphyto}{dt} = phyto \cdot 0.55 \cdot \frac{ps}{ps+0.033} \cdot \frac{silica^2}{silica^2+0.02} \cdot \frac{no}{no+0.073} \cdot 1.11^{(temp-18.5)} \cdot \frac{light}{140} \cdot e^{\left(1-\frac{light}{174.9}\right)} -$$

$$- phyto \cdot 0.092 \cdot \frac{temp-0.4}{19.4-1.5} - phyto \cdot \frac{0.09}{10} \cdot 1.11^{(temp-15.3)} - daph \cdot 10^{-5} \cdot \frac{temp}{15-3.4} \cdot \frac{phyto}{phyto+2.1} \cdot phyto \cdot 0.37$$

19

2000:

$$\frac{dphyto}{dt} = phyto \cdot 1.9 \cdot \frac{ps}{ps+0.0017} \cdot (1-e^{-0.11 \cdot silica}) \cdot \frac{no^2}{no^2+0.012} \cdot 1.13^{(temp-15)} \cdot \frac{light}{100} \cdot e^{\left(1-\frac{light}{116.3}\right)} -$$

$$- phyto^2 \cdot 0.004 \cdot \frac{temp}{5.2} - phyto \cdot \frac{0.23}{10} - daph \cdot 0.52 \cdot \frac{temp}{9.1} \cdot \frac{phyto}{phyto+3.9} \cdot phyto \cdot 0.72$$

20

2001:

$$\frac{dphyto}{dt} = phyto \cdot 9.3 \cdot \frac{ps^2}{ps^2 + 0.09} \cdot (1 - e^{-5.15 \cdot silica}) \cdot \frac{no}{no + 0} \cdot 1.13^{(temp-15)} \cdot \frac{light}{114.4} \cdot \exp\left(1 - \frac{light}{197.9}\right) -$$

$$- phyto^2 \cdot 0.0005 \cdot \frac{temp}{2.4} - phyto \cdot \frac{0.5}{10} \cdot \frac{temp - 4.7}{15 - 5} - daph \cdot 0.33 \cdot \frac{temp}{15 - 5} \cdot \frac{phyto}{phyto + 0.26} \cdot phyto \cdot 0$$

21

2002:

$$\frac{dphyto}{dt} = phyto \cdot 9.4 \cdot \frac{ps}{ps + 0} \cdot \frac{silica^2}{silica^2 + 15} \cdot \frac{no^2}{no^2 + 10} \cdot \frac{temp}{5.7} \cdot \frac{light}{light + 41} - phyto^2 \cdot 0.0054 \cdot \frac{temp}{5.1}$$

$$- phyto \cdot \frac{0.05}{10} - daph \cdot 10^{-5} \cdot \frac{temp - 2}{19.8 - 2.6} \cdot \frac{phyto}{phyto + 17.3} \cdot phyto \cdot 0.15$$

22