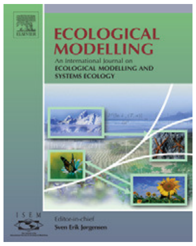
available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/ecolmodel



EcoNet: A new software for ecological modeling, simulation and network analysis

Caner Kazancı

University of Georgia, Department of Mathematics and Faculty of Engineering, Athens, GA 30602, United States

ARTICLE INFO

Article history:
Published on line 9 July 2007

Keywords:
EcoNet
Ecological modeling
Network analysis
Ecological simulation
Flow analysis
Food web
Software

ABSTRACT

We present EcoNet, a simulation and analysis software for ecological systems. EcoNet integrates dynamic simulation capability with steady-state network analysis. It features a simple and flexible interface, offering a gentle learning curve for the novice modeler. EcoNet contains sophisticated numerical equation solving routines, which require very few user choices, bringing the thought process and simulation results closer. Features like automatic diagram creation, capability to model large systems with thousands of stocks and flows, stochastic simulation options, and a web interface that requires no installation make EcoNet a welcome addition to the ecological modeling world.

1. Introduction

Recent advances in high throughput experimental techniques have generated large amounts of data on biological systems, cellular pathways and genetic networks. For example, the amount of nucleotide sequence accumulated in the GenBank in the last 3 years is more than half the total data accumulated since its establishment in 1982 (NCBI, 2006). Flow of new information led to the realization of the complex structure and behavior of biological systems.

A major reason contributing to the complexity of these systems is the *network structure* with many interactions among multiple identities. In a genetic network, these identities are genes, and the interactions among genes are up-regulation and down-regulation. In a cellular pathway, identities are molecular species, and interactions are biochemical reactions. In ecological systems, identities can range from accumulated organic matter to hundreds of species, interactions may represent flow of energy, biomass or a specific element such as C, N or P.

A common way to simulate these systems is to form a set of differential equations where the solution represents the state of each identity changing in time. There are over 30 software (Ramsey et al., 2005) specifically developed to analyze biological systems and Ecosim (Gotelli and Entsminger, 2006) is one of the few designed specifically for ecological models. Using these software, a simplified version of the real system is created and then calibrated with available data. A functioning model will provide insights as to how the real system works, how it can be controlled and manipulated.

Another way to analyze these systems is by formulating system-wide organizational properties. Consider an ecological model where the state of each identity, and the flows among identities are steady. A differential equation based simulation, since it only predicts future dynamic behavior of each individual identity, will not provide any insights as to how the environmental inputs are shared among identities, how much energy or matter cycling occurs within the system, or how strong are any two identities in the system related to each other. Obviously, such analysis is essential in understanding how a specific ecological system functions, how it can be sustained or manipulated.

EcoPath (Christensen et al., 2002) is a powerful software that computes such system-wide properties. Initially devel-

---

E-mail address: caner@uga.edu.

0304-3800/$ — see front matter. Published by Elsevier B.V.
doi:10.1016/j.ecolmodel.2007.04.031

oped as separate software, EcoPath and EcoSim are combined in one software package and provide both dynamic simulation and network analysis. The package includes many other features such as EcoSpace, a module that enables spatial modeling. Similar, but not as comprehensive, EcoNet also features dynamic simulations and steady-state network analysis.

EcoNet is designed to simplify the model building, simulation and analysis effort, and its power lies in its simplicity. EcoNet runs on a server at http://eco.engr.uga.edu and requires no installation. Instead, users enter their models through a web browser, in an intuitive text-based format. The model is then automatically converted into differential equations and solved numerically. Network analysis is performed based on the final state of the solution, and results are fed back to the user's browser along with a nice network diagram of their model. Typically, this process takes less than a second. Any process that can be represented as a stock-flow diagram can be implemented in EcoNet within a few minutes. The simplicity of EcoNet interface not only encourages first time modelers to access a powerful modeling tool, but also minimizes the model building effort for experienced users, closing the gap between the thought process and the results.

Designing ecological modeling software with a gentle learning curve for the novice user, and powerful features for the demanding expert user is a challenging task. To combine these two contradicting design perspectives, we hid sophisticated mathematical techniques and efficient numerical methods behind a simple and flexible user interface. The novice user can run EcoNet without seeing a single differential equation, while an expert user can exploit EcoNet's high-end features for demanding applications. For example, EcoNet features a very recent and sophisticated stochastic simulation algorithm (Gillespie, 2000), which is easily accessed by opting for the *stochastic method*. It is even possible to run automatically generated large spatial models because the numerical engine of EcoNet can handle models that contain up to 10^5 stocks and 10^5 flows.

2. User interface

EcoNet runs on a server through a web interface. EcoNet web-page (http://eco.engr.uga.edu) needs three types of inputs from the user: the model written in text format, a choice of numerical method and numerical method parameters. After entering this data on EcoNet web-page, the user clicks on "`Run Model`" to retrieve the results. Conveniently, a simple model is displayed and default options for numerical methods and parameters are pre-selected. Anyone can try and run EcoNet immediately without needing any prior knowledge.

During the design process of EcoNet interface, we had several choices for a model input method. A widely used dynamic simulation software Stella (Clauset et al., 1987) uses a graphical user interface where users drag and drop geometrical shapes representing stocks and then define arrows connecting these stocks. EcoNet uses a flexible text-based model input method. For example, to represent the simple model given by the diagram in Fig. 1, user simply types the following:

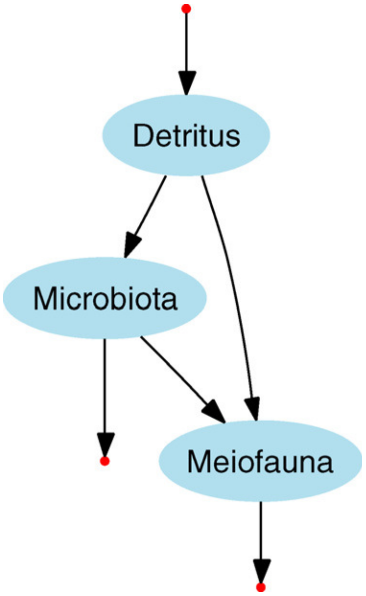


Fig. 1 – A simple model diagram, created automatically by EcoNet.

```
* -> Detritus
Detritus -> Microbiota
Detritus -> Meiofauna
Microbiota -> Meiofauna
Meiofauna -> *
Microbiota -> *
```

While a GUI-interface is intuitive and user friendly, it is not necessarily easier to use. Typing a text file is considerably quicker and easier than using a mouse to form a diagram.

A model in text format is extremely portable and does not need special file formats to be saved or sent. It can be copied and pasted into other applications, and is human readable. Furthermore, a text-based interface enables expert users to write codes to automatically generate very large models with thousands of stocks and flows for spatial modeling or statistical analysis.

Visualization is a strong aspect of a GUI-interface. However, for slightly larger systems, a model built with a GUI-interface contains too many flow lines crossing each other, creating a messy visual diagram. It becomes harder to build the model, and the final diagram does not provide much visual information. A useful diagram should provide insight on system structure and behavior by locating important stocks with many connections in the center, while keeping stocks with fewer connections closer to the edges. Flow lines should be short with little curvature, and should not intersect often. Unfortunately, this is not only a difficult task for the user, but a difficult optimization problem for the software engineer.

Therefore, we designed EcoNet to create an optimized diagram using only the text-based model input. EcoNet uses Graphviz (Ellson et al., 2003), a professional open source graph visualization software, to determine the optimum placement of stocks and flow lines to achieve the listed goals. The diagram also reflects the trophic level of the compartments, even in the case of mixed trophic levels with cross level interactions.

3. Simulation

Simulations are handled by two different modules in EcoNet. First module automatically converts the user's model input into a differential equation system. Second module numerically solves this differential equation using the selected method and parameters. First module needs the following information to create the differential equation:

(1) Flows between compartments.
(2) Initial value of compartments.
(3) "Speed" of flows.

Here is a complete model corresponding to the diagram given in Fig. 1:

```
* -> Detritus            c=10
Detritus -> Microbiota   c=.15
Detritus -> Meiofauna    c=.2
Microbiota -> Meiofauna  c=.5
Meiofauna -> *      c=.23
Microbiota -> *     c=.01
Detritus = 100, Microbiota = 50
Meiofauna = 10
```

Here, the numbers following "c=" are called flow coefficients, and are proportional to the speed of the flow. "*" represents the environment. The initial values are given in the last two lines of the model input. It seems that we implicitly suggest a model writing structure in this example. Each flow is written on a separate line, followed by its flow coefficient, and initial conditions are written at the end. To the contrary, EcoNet is extremely flexible in model interpreting, and will gladly accept the following as a valid model input and generate exactly the same results as the previous version:

```
Detritus = 100
Microbiota -> Meiofauna; Detritus -> Meiofauna
c=0.5, c=.2; c=10      #comments...
Microbiota = 50, Meiofauna = 10
* -> Detritus; Detritus -> Microbiota   c=0.15
Meiofauna -> *  c=.23; Microbiota -> *   c=.01
```

text-based human–computer interactions are generally very demanding on the user's side. Sometimes, even blank space must be accounted for when writing a computer code. However EcoNet users can write their models with a great range of flexibility, as EcoNet does not assume any order or formatting, and does not use any specific words as identifiers. EcoNet recognizes each compartment name, and classifies each user input as an initial condition, a flow or a flow coefficient. It is EcoNet that does the hard work, not the user.

Here are some guidelines on how EcoNet reads your model: EcoNet does not distinguish among "`;`", "`,`", "`Tab`" and the "`New Line (Enter)`" character. In general, EcoNet does not take order of appearance into consideration. Only when the flow coefficients are grouped separately from the flows, as in our second model version, each flow coefficient is associated to a flow in the order of appearance. Consecutive spaces and empty lines are ignored. Users can add comments anywhere in their model starting with "#" character.

While flexibility is a great feature, it also becomes easier to make mistakes when writing model in a disorganized manner. So we equipped EcoNet with a good error tracking feature; that generates meaningful error messages when there are mistakes in the model. For example, misspelling "`Meiofauna`" when defining the initial condition will receive the following complaint from EcoNet: "`No initial condition exists for node Meiofauna`". EcoNet models are case-sensitive.

After EcoNet properly recognizes the model, it is then converted into a differential equation system. For example, the differential equations for the simple model given above are as follows:

$$\frac{d[\text{Detritus}]}{dt} = 10 - 0.15\,[\text{Detritus}] - 0.2\,[\text{Detritus}]$$

$$\frac{d[\text{Microbiota}]}{dt} = 0.15\,[\text{Detritus}] - 0.5\,[\text{Microbiota}] \\ -0.01\,[\text{Microbiota}]$$

$$\frac{d[\text{Meiofauna}]}{dt} = 0.2\,[\text{Detritus}] + 0.5\,[\text{Microbiota}] \\ -0.23\,[\text{Meiofauna}]$$

EcoNet uses donor controlled mass-action kinetics, that is, the rate of a flow is computed as the product of the flow coefficient and the stock value of the originating compartment. The speed of the flow

$$A \rightarrow B \quad c = 2.5$$

is computed as

$$(\text{speed of flow } A \rightarrow B) = 2.5 \times [A]$$

We are currently working to incorporate other type of kinetics, like Lotka-Volterra (Matsuda et al., 1992) and Michaelis-Menten (Dowd and Riggs, 1965) into EcoNet. Lotka-Volterra type kinetics is useful when a flow speed depends on both the donor and the recipient stock value, which is the case with most food webs. Michaelis-Menten kinetics, originally developed for enzymatic reactions, is useful when a flow is limited or mediated by other factors. While enriching EcoNet with these features, we will preserve the simplicity of EcoNet interface.

The syntax for defining a flow with Michaelis-Menten kinetics will be as follows:

A → B v = 2.5,1.5

The speed of this flow will be interpreted as:

$$\text{(speed of flow } A \rightarrow B) = \frac{2.5 \times [A]}{1.5 + [A]}$$

EcoNet will automatically recognize the type of kinetics based on the letters used to express the flow coefficients. Similarly, we use the letter `r` to specify Lotka-Volterra kinetics. This new syntax keeps the EcoNet interface simple, and is backward compatible with the current version. Novice users can still use simpler mass-action kinetics without getting confused, while expert users can utilize this new future with a simple switch from the letter “`c`” to “`r`”.

4. Numerical methods

Currently, EcoNet offers four numerical methods for the solution of the differential equation system:

1. Fourth order Runge-Kutta (fixed step-size)
2. Runge-Kutta-Fehlberg (adaptive step-size)
3. Fast stochastic (based on Langevin equation)
4. Discrete stochastic algorithm (based on Gillespie’s Algorithm)

Fourth order Runge-Kutta method is offered as the only deterministic solver using fixed time-steps. A simpler solver, Euler method, is a part of the EcoNet numerical engine. However, it is not offered as an option in the web interface simply because it has no advantage over the fourth order Runge-Kutta method. In general, more accurate solvers are more complex and therefore require less iterations but more computing time at each iteration. Fourth order Runge-Kutta algorithm, provides an optimal balance between complexity and efficiency.

Fixed step-size solvers are based on the assumption that the actual solution does not change significantly during a short time interval. In most simulations, system state changes fast initially and then slowly converges to a steady-state. Adaptive methods conform to the differential equation, and use small step-sizes when the solution changes rapidly, and switch to larger step-sizes where the solution is smoother. Therefore, these solvers use a `tolerance` parameter, instead of `step-size`. While providing great efficiency, adaptive methods are not perfect; so fixed step-size methods are always offered as a safer option.

Stochastic solvers take the probabilistic system behavior into account and generate different solutions at each run. It seems that stochastic solvers are producing random results. In most cases, they are more accurate because no real biological or ecological system is deterministic in nature. It only makes sense to use a stochastic solver for a system which is stochastic in nature. However, few software feature such solvers, mainly because they are complex and hard to implement. Since they are not widely available, few users are aware of their power. For example, a single stochastic solution will reveal the inherent variations in the stock values, eliminating the need for many runs for sensitivity analysis.

EcoNet features a very fast stochastic method based on the Langevin equation (Gillespie, 2000). Published in 2000, this fairly sophisticated second order solver is based on a Fokker-Planck type partial differential equation (PDE) derived from a discrete stochastic process, which is then converted to a stochastic differential equation (SDE). EcoNet also features a completely different stochastic solver based on Gillespie’s stochastic algorithm (Gillespie, 1977). This interesting method uses integers rather than real numbers, and provides Individual-based-model (or agent-based-model) flexibility and deterministic model compatibility. We refer the discussion of this interesting method to another paper (Kazancı and Tollner, in preparation).

EcoNet creates a figure that shows the temporal changes of all stock values over time. When using a stochastic method, this figure will be different at each simulation run even if all the parameters are the same. Fig. 2 shows these figures for both a deterministic and a stochastic method run. These plots provide information as to how the system evolves from the

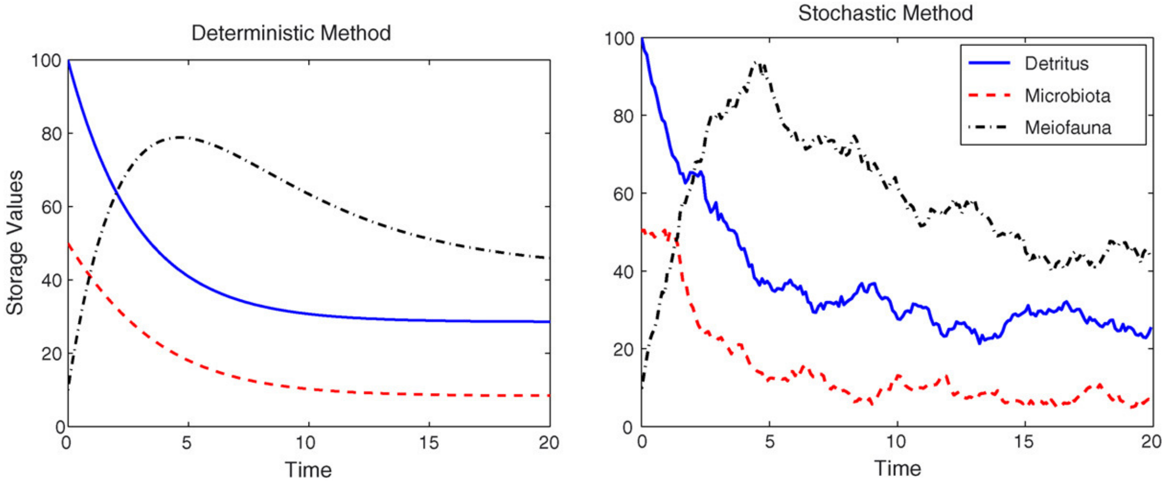


Fig. 2 – EcoNet automatically creates figures of temporal evolution of stock values over time. Two different runs for the simple model are shown. The first graph uses fourth order Runge-Kutta method while the second one uses the fast stochastic algorithm.

given initial condition, and also informs the user if the system reaches steady-state or not.

5. Network analysis

EcoNet uses Network Environ Analysis (NEA). The heart of the NEA theory is the fact that the relation between two stocks does not only depend on the direct flow between them, but mostly depend on the indirect flows involving other compartments. Unlike most analysis methods, NEA treats the system as a whole and provides an elegant way to quantify the effects of indirect flows in the system. NEA is not a one-step analysis, but a series of algebraic operations resulting in scalar, vector and matrix values representing various system-wide properties of ecological systems.

EcoNet analysis results include the adjacency matrix (A), community matrix (C), steady-state flow matrix (F), dimensionless flow matrix (B), through-flow analysis (N), storage analysis (S) and utility analysis (U). Adjacency matrix consists of 0 and 1 entries and indicates if there exists a direct flow between two stocks. In other words, $A_{ij} = 1$ if there is a direct flow from stock j to stock i, $A_{ij} = 0$ otherwise. Flow coefficients form the community matrix. C_{ij} is the flow coefficient from stock j to stock i.

Adjacency and community matrices are formed by the model input only, so they do not depend on the simulation results. The rest of the network analysis rely on the steady-state values achieved by the simulation results, and will not be accurate unless the system reaches steady-state. EcoNet performs network environ analysis based on the final state reached by the simulation run, whether it is close to steady-state or not. The user should judge how close the system is to steady-state by viewing the figure showing the time course of stock values.

The flow matrix F shows the amount of energy or mass flow per time units between compartments at steady-state. Assuming mass-action kinetics

$$F_{ij} = C_{ij} X_j$$

where X_j denotes the value of stock j at steady-state. S_{ij} represents how much of the steady-state value of stock i is contributed by the environment input to stock j. Note that there might be many possible paths involving other stocks for an environment input to reach stock i, and S_{ij} accounts for all such possible paths. Mathematically, storage analysis matrix (S) provides a linear mapping from environment inputs to stock values:

$$S : \bar{z} \mapsto \bar{X} \quad \bar{X} = S\bar{z}$$

Here, $\bar{X}$ represents a vector of all steady-state stock values, and $\bar{z}$ represents a vector where $\bar{z}_i$ is the coefficient of flow from the environment to stock i.

Through-flow of stock i is the total amount of mass or energy received by that stock per unit time at steady-state, and is given by

$$T_i = \sum_j F_{ij}$$

Similar to storage analysis, through-flow analysis (N) provides a mapping from environment inputs to through-flow values:

$$N : \bar{z} \mapsto \bar{T} \quad \bar{T} = N\bar{z}$$

where $\bar{T}$ represents a vector of through-flow values. Assuming the through-flow value of a stock indicates the amount activity at that stock, N_{ij} represents how much of the activity at stock i is contributed by the environment input to stock j.

Utility analysis is a rather involved part of NEA. It is a powerful method that provides "relations" among stocks, again including indirect effects. In the following example

$$* \rightarrow \text{tree} \rightarrow \text{deer} \rightarrow \text{wolf} \rightarrow *$$

deer and *wolf* have a $(-, +)$ relationship, same as the *tree–deer* relationship. Although not connected with a direct flow, the *tree* and the *wolf* have a mutualistic $(+, +)$ relationship. Figuring out such relations is straightforward for simple models, but extremely difficult when models involve feedback loops, cycling or cross-level feeding. Utility analysis provides this relation among all stocks regardless of model complexity. For further details on utility analysis, and NEA in general, we refer the reader to (Fath and Patten, 1999; Patten, 1978, 1999; Gattie et al., 2005).

6. History of EcoNet

Although EcoNet software is less than a year old, we initially developed its numerical engine back in 2001 at Carnegie Mellon University to analyze statistical properties of large biochemical networks. Failing to find software able to simulate biochemical networks involving over 10,000 molecules and reactions, we developed ours from scratch in C++. We even wrote our own optimized linear algebra libraries as most available ones performed poorly for large systems. After 4 years of experience with this code, we have eliminated most bugs and created a robust and refined code, which EcoNet is based on.

We wrote new codes for network analysis, diagram generation, time course plotting and an interpreter enabling flexible user input. The modular structure of EcoNet enables us to extend its capability by adding new features with minimal effort. While the minimalistic user interface of EcoNet looks similar to many web applications (Java applets) that run on the client side, EcoNet runs on the server side and is equipped with a numerical engine far superior to many commercial software. When a user clicks on "`Run Model`" on EcoNet webpage, the server receives and evaluates the submitted model. Unless errors are detected in the model, a sequence of C++ codes and unix shell scripts work together to generate the simulation and analysis results. The server then creates a new web-page that contains these results, which is then loaded into the user's browser. Typically, this process takes less than a second. EcoNet currently runs on a powerful multi-processor Linux workstation-server.

7. Future work

EcoNet was made available on-line on June 2006, and is still at its infancy. We are continuously improving EcoNet while keeping the interface simple and clean. Our goals are to improve the user interface, add new numerical techniques and analysis methods, and increase its efficiency and flexibility.

A major improvement is the addition of Lotka-Volterra and Michaelis-Menten type flows which we discussed earlier. In near future, EcoNet analysis results will include ascendancy (Ulanowicz, 1997), exergy (Jorgensen and Svirezhev, 2004) and cycling index (Finn, 1976), making it an invaluable tool to compute and compare various network properties with minimal effort (Patten, 1995; Jorgensen, 1994).

A recent extension of EcoNet is the "Particle Tracking Algorithm" (Kazancı and Tollner, in preparation), which provides a Lagrangian approach to analyze network flows with amazing detail. Using the same interface as EcoNet, this new method enables us to search for new properties, verify the current network properties and investigate ecological thermodynamics (Tollner and Kazancı, submitted for publication).

It is generally hard to establish the steady-state of a real system solely based on field data, and simulation may be necessary before the network analysis study. EcoNet combines dynamical simulations with steady-state network analysis. Therefore, any research on steady-state network analysis (Patten, 1992; Gattie et al., 2006; Schramski et al., 2006; Borrett and Osidele, 2007) would benefit from EcoNet. Furthermore, simple interface of EcoNet enables users to quickly build models and run simulations without even writing specific differential equations. Research on dynamical simulations of ecological systems and food webs (Salles and Bredeweg, 2006; Hearne and Swart, 1991), especially ones that contain perturbation or sensitivity analysis would certainly benefit from EcoNet.

Acknowledgements

I would like to thank D. Gattie, N. Kellam, B. C. Patten, J. Schramski, E. W. Tollner, S. J. Whipple and the Systems and Engineering Ecology group at the University of Georgia for their help and guidance in the development of EcoNet. Special thanks go to S. Ta'asan at Carnegie Mellon University for his support in the development of the numerical engine. Creating EcoNet would be very hard without the technical help of F. Monalache at CMU. Finally, I would like to thank the students in ECOL 8580 for their valuable feedback.

REFERENCES

Borrett, S.R., Osidele, O.O., 2007. Environ ndicator sensitivity to flux uncertainty in a phosphorus model of Lake Sidney Lanier, USA. Ecol. Model. 200, 371–383.

Christensen, V., Walters, C.J., Pauly, D., 2002. Ecopath with ecosim version 5. Univ. of British Columbia, Fisheries Centre, Vancouver, Canada.

Clauset, K.H., Rawley, C.C., Bodeker Jr., G.C., 1987. Stella—software for structural thinking. Collegiate Microcomput. 5 (4), 311–319.

Dowd, J.E., Riggs, D.S., 1965. A comparison of estimates of michaelis-menten kinetic constants from various linear transformations. J. Biol. Chem. 240, 863–869.

Ellson, J., Gansner, E.R., Koutsofios, E., North, S.C., Woodhull, G., 2003. Graphviz and dynagraph—static and dynamic graph drawing tools. In: Junger, M., Mutzel, P. (Eds.), Graph Drawing Software. Springer-Verlag, pp. 127–148.

Fath, B.D., Patten, B.C., 1999. Review of the foundations of network environ analysis. Ecosystems 2, 167.

Finn, J.T., 1976. Measures of ecosystem structure and function derived from analysis of flows. J. Theor. Biol. 56, 363–380.

Gattie, D.K., Schramski, J.R., Borrett, S.R., Patten, B.C., Bata, S.A., Whipple, S.J., 2006. Indirect effects and distributed control in ecosystems Network environ analysis of a seven-compartment model of nitrogen flow in the Neuse River Estuary, USA—steady-state analysis. Ecol. Model. 194, 162–177.

Gattie, D.K., Tollner, E.W., Foutz, T.L., 2005. Network environ analysis: a mathematical basis for describing indirect effects in ecosystems. Trans. ASAE 48 (4), 1645–1652.

Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81, 2340.

Gillespie, D.T., 2000. The chemical langevin equation. J. Chem. Phys. 113, 297.

Gotelli, N.J., Entsminger, G.L., 2006. Ecosim: null models software for ecology.

Hearne, J.W., Swart, J., 1991. Optimal translocation strategies for saving the Black Rhino. Ecol. Model 59, 279–292.

Jorgensen, S.E., 1994. Review and comparison of goal functions in systems ecology. Vie Milieu 44, 11–20.

Jorgensen, S.E., Svirezhev, Y.M., 2004. Towards a Thermodynamic Theory for Ecological Systems, first ed. Elsevier.

Kazancı, C., Tollner, E.W. Particle tracking: a Langangian approach to dynamic ecological network analysis. Ecol. Model., in preparation.

Matsuda, H., Ogita, N., Sasaki, A., Sato, K., Dec 1992. Statistical mechanics of population: the lattice Lotka-Volterra model. Prog. Theor. Phys. 88, 1035–1049.

NCBI Growth of GenBank, 2006. Date of publication: March 7, 2006. Date retrieved: December 3, 2006. http://www.interaction-design.org/references/.

Patten, B.C., 1978. Systems approach to the concept of environment. Ohio Acad. Sci. 78 (4), 206–222.

Patten, B.C., 1992. Energy, emergy and environs. Ecol. Model 62, 29–69.

Patten, B.C., 1995. Network integration of ecological extremal principles: exergy, emergy, power, ascendancy, and indirect effects. Ecol. Model. 79, 75–84.

Patten, B.C., 1999. Holoecology: The Unification of nature by Network Indirect Effects. Kluwer (Unpublished).

Ramsey, S., Orrell, D., Bolouri, H., April 2005. Dizzy: stochastic simulation of large-scale genetic regulatory networks. J. Bioinform. Comput. Biol. 3 (2), 41.

Salles, P., Bredeweg, B., 2006. Modelling population and community dynamics with quantitative reasoning. Ecol. Model 195, 114–128.

Schramski, J.R., Gattie, D.K., Patten, B.C., Borrett, S.R., Thomas, C.R., Whipple, S.J., 2006. Indirect effects and distributed control in ecosystems: Distributed control in the environ networks of a seven-compartment model of nitrogen flow in the Neuse River Estuary, USA - Steady-state analysis. Ecol. Model 194, 162–177.

Tollner, E.W., Kazancı, C. Discrete simulation as a basis for connecting network environ analyses to ecological thermodynamic theory. Ecol. Model., submitted for publication.

Ulanowicz, R.E., 1997. Ecology, the Ascendent Perspective, first ed., first edition. Columbia University Press.