ISSN: 2167-1435
v30 n2536

Issues of Digital Ecosystems Approach to Biological Ecosystem Simulation Model Fit Optimization

Justin C. W. Debuse
jdebuse@usc.edu.au
Faculty of Arts and Business
University of the Sunshine Coast,
Queensland, Australia

Shah J. Miah
Shah.Miah@vu.edu.au
College of Business
Footscray Park Campus,
Victoria University, Victoria, Australia

V. Debuse
valerie.debuse@daff.qld.gov.au

S. Lawson
simon.lawson@daff.qld.gov.au

Queensland Government Department of Agriculture, Fisheries and Forestry,
Brisbane, Queensland, Australia

Abstract

This study examines the application of digital ecosystems concepts to a biological ecosystem simulation problem. The problem involves the use of a digital ecosystem agent to optimize the accuracy of a second digital ecosystem agent, the biological ecosystem simulation. The study also incorporates social ecosystems, with a technological solution design subsystem communicating with a science subsystem and simulation software developer subsystem to determine key characteristics of the biological ecosystem simulation. The findings show similarities between the issues involved in digital ecosystem collaboration and those occurring when digital ecosystems interact with biological ecosystems. The results also suggest that even precise semantic descriptions and comprehensive ontologies may be insufficient to describe agents in enough detail for use within digital ecosystems, and a number of solutions to this problem are proposed.

Keywords: digital ecosystems, simulation, optimization, ontologies

1. INTRODUCTION AND MOTIVATION

This study builds upon existing digital ecosystems research (Debuse & Miah, 2011) to examine the application of digital ecosystems principles (Boley & Chang, 2007) to define a simulation optimization problem. The problem focuses upon improving the quality of a mathematical model of a biological system. This study further examines the progress made towards this goal along with challenges faced, within the context of digital ecosystems.

Digital ecosystems principles offer a number of potential benefits to this study. Animals, software and humans can all be analyzed in a similar manner, namely as agents, and interactions between overlapping ecosystems can be examined (Boley & Chang, 2007). Digital ecosystems principles may be applied to the analysis of all agents, such as the precision of semantic descriptions together with mapping and translation between rules and ontologies that are not fully compatible (Boley & Chang, 2007).

Existing research shows support for enhancement of optimization processes and techniques across human oriented technological solution designs. For example, digital ecosystems has been used for support and foundation of optimization of electricity market bidding, including technologies such as swarm intelligence (Zhang, Gao, & Lu, 2011). The PolyWorld system applies digital ecosystems concepts much more broadly, with the goal of incorporating all key elements of living systems such as metabolisms, physiologies and genetics into a single artificial system (Yaeger, 1994). Disciplines such as neurophysiology, behavioral ecology and evolutionary biology can use PolyWorld as a research tool (Yaeger, 1994); for example, PolyWorld has been used to examine evolutionary selection of brain network topologies (Yaeger, Sporns, Williams, Shuai, & Dougherty, 2010). Motivated by such research, and examining an application within the human systems domain using the digital ecosystems paradigm, the study begins with a brief description of the original problem scenario, before introducing its current state; the digital ecosystems, social ecosystems and biological ecosystems perspectives (Boley & Chang, 2007) are all discussed.

Next, the results, including progress made and challenges faced, are presented, before the discussion is given and conclusions are drawn regarding the implications for future digital ecosystems research and practice.

2. PROBLEM SCENARIO

The original problem scenario upon which this study builds was presented in (Debuse & Miah, 2011), within which the digital ecosystem contains two agents (Boley & Chang, 2007). The first agent is for the management of forest pests, and is a mathematical model of forest insect pest species (Nahrung, Schutze, Clarke, Duffy, Dunlop, & Lawson, 2008); this agent effectively models part of a biological ecosystem. Although the agent has been found to be robust in terms of how accurately it models the biological ecosystem (Nahrung et al., 2008), it may potentially be improved by optimizing its parameter values so that it fits the data more closely (Debuse & Miah, 2011). The second agent is an optimization algorithm, such as a genetic algorithm, which modifies the parameter values of the first agent to improve the quality of its fit to the data (Debuse & Miah, 2011). The second agent cannot use a more simplistic approach such as brute force search, due to the enormous number of combinations of parameter values that exist.

The proposed digital ecosystem formed part of a social ecosystem comprising a number of subsystems and agents (Boley & Chang, 2007). The science subsystem contained scientists acting as agents to develop and test the optimization mode. For example, the simulation parameters to be optimized must be selected, along with constraints to ensure that they are biologically realistic (Debuse & Miah, 2011). The technological solution design (TSD) subsystem comprised IT researchers acting as agents to apply digital ecosystems principles and develop the optimization approaches (Debuse & Miah, 2011). This subsystem was thus responsible for tasks such as determining the most appropriate optimization algorithms and problem representation for the simulation optimization problem (Debuse & Miah, 2011).

Within the digital ecosystem of the current problem scenario, the first agent, which performs the simulation task, has been created within the DYMEX (Maywald, Bottomley, & Sutherst, 2007) modeling environment. DYMEX allows process based biological simulation models to be developed without users requiring programming expertise (Maywald et al., 2007), and thus allows representations of real biological ecosystems to be created as digital ecosystems. The next step is to create the second agent within this ecosystem to optimize the parameters of this first agent. From a social ecosystem perspective, this requires collaboration between the science and TSD subsystems. Further, the goal is to ensure that the simulation agent models the real biological ecosystem as accurately as possible. This mirrors the collaboration between digital ecosystems described in (Boley & Chang, 2007); a digital ecosystem is collaborating with a biological system. Similar challenges are present. In digital ecosystem collaborations, mapping and translation may be required between digital ontologies and rules that are not fully compatible (Boley & Chang, 2007); similarly, the optimization agent in this study is responsible for improving the quality of mapping and translation from the biological ecosystem to its digital counterpart.

3. RESEARCH METHOD AND RESULTS

Social Ecosystems

Social ecosystems reinforce collaborative relationships between software and social agents to enhance operational efficiency. The social ecosystems needed to be expanded to include developers of the simulation software as a third subsystem, to determine how the simulation and optimization agents within the digital ecosystem could communicate and collaborate. Figure 1 shows the social ecosystems in this study.

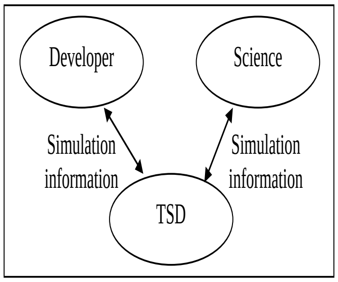

Figure 1. Social ecosystems within this study

The DYMEX simulation environment was reported by the developer subsystem to provide support for a brute force optimization based agent. The exchange of messages between DYMEX and external agents (Boley & Chang, 2007) would allow an external second agent to perform the parameter optimization, but this appeared to be unsupported.

The solution adopted was thus for the researchers within the TSD subsystem to recreate the DYMEX model within an environment which allowed interaction with the EVA2 (Evolutionary Algorithms framework version 2) framework proposed to support the optimization agent (Debuse & Miah, 2011). EVA2 uses Java to support optimization algorithms such as genetic algorithms (Kronfeld, Planatscher, & Zell, 2010), and so the simulation was recreated as Java software.

Despite the relative simplicity of the simulation, the recreation of the simulation required collaboration between the TSD, simulation software developers and science subsystems. This initially involved message exchange between the TSD and science subsystems to determine operation of the simulation. A key challenge was the ontological complexity of the simulation software, along with the extent to which this appeared to be made fully explicit within the system. Ontology provides a formal structure in which an explicit specification of the system can be presented. Previous studies identify the importance of ontology development in knowledge engineering, especially for knowledge integration (Chen, 2010; Soo, Lin, Yang, Lin, & Cheng, 2006). In knowledge based system design, a recent study by Chen (2010) identified knowledge management issues inside an enterprise. To address this, Chen (2010) developed an ontology based approach to knowledge reasoning. This approach mainly utilizes a web ontology language (called WOL) to represent empirical knowledge in a structured manner to enable sharing and reuse. In our system design, we used the ontology to determine how the Java version of the simulation should be constructed. The ontology provided both guidance for appropriate class structures within Java and details of how the simulation software functioned, so that this could be accurately recreated within Java.

As development of the Java version of the simulation proceeded, further message exchange was required to determine the potential cause of discrepancies between the Java and DYMEX versions of the simulation, with the developer subsystem being involved due to their greater knowledge of the DYMEX system.

Digital Ecosystems

Simulation agent
Development of the Java version of the DYMEX implementation of the simulation agent presented a number of challenges.

Firstly, despite the model containing only seven stages, from egg through to overwintering adult, its overall complexity made it challenging to model; Table 1 contains the pseudocode. For every simulated day of forest pest insect development, each simulation life stage was run. Each stage contained five key functions: chronological age; development, determining how individuals age physiologically; mortality, determining how individuals die; stage transfer, controlling how individuals move from one stage, such as from egg to larva; and progeny generation, calculating how new eggs are produced. Many of the functions contained multiple parameters; further, some stages contained additional complexity, such as having dual stage transfer or including reproduction. The length and temperature of each day were also used within the simulation.

The simulation also used the concept of cohorts, representing a group of individuals that are created at the same time and share characteristics such as their physiological age. Multiple cohorts were implemented; each stage of the simulation could thus potentially contain a number of different cohorts.

The original DYMEX simulation possessed digital ecosystem entity characteristics such as precise semantic description and a comprehensive ontology (Boley & Chang, 2007). An ontology can act as a vocabulary of a digital ecosystem, grouping properties and categories hierarchically (Boley & Chang, 2007). Thus, the ontology of the DYMEX simulation included categories such as life stages and functions; these were recreated within the Java version. The power and flexibility of the Java language should avoid the potential for problems relating to the lack of full compatibility between the ontologies and rules of the two systems (Boley & Chang, 2007); Java should be able to emulate any DYMEX computation. However, differences between the two simulations still emerged.

Tracing the source of the differences proved challenging, since the precision to which DYMEX calculated many of the results was higher than that to which they were reported; determining the exact nature of its operation was impossible in some areas. Experimenting with multiple scenarios within the Java system allowed the sources of some discrepancies to be identified; for others, the range of potential scenarios to investigate was too large to be explored exhaustively, and required input from the developer social subsystem.

```
for each day
  for each life stage
    for each cohort
      compute chronological age
      compute physiological age
      apply continuous mortality
    for each stage transfer function
    $n_t$ := 0
      for each cohort
        $n_t$ := $n_t$ + computed number of cohort to transfer to next stage
        remove transferred individuals from cohort
      compute and apply reduction in $n_t$ caused by exit mortality
      $c_{n1}$ := new cohort containing $n_t$ individuals
      add $c_{n1}$ to the next life stage
    $n_p$ := 0
    for each cohort
      compute progeny generation rate $g_r$
      $n_p$ := $n_p$ + progeny computed using cohort size, $g_r$ and fecundity
    compute updated fecundity
    $c_{n2}$ := new cohort containing $n_p$ individuals
    add $c_{n2}$ to the egg life stage
```

Table 1. Java simulation model pseudocode.

Differences in precision were also responsible for differences between the simulations. When DYMEX tested whether a cohort was empty it rounded its size before applying the test. Such a detail appeared not to be described within the documentation for the system, and would be unlikely to be of interest to a scientist building or testing models due to its subtle effect; however, this caused divergence between the simulations after a number of simulated days, which was reduced when the Java version was modified to use the same rule.

The simulation was run for over 200 simulated days in total, which introduced further scope for divergence between the two simulations, since even very small initial discrepancies could escalate over time into much larger gaps. Further research was required before the two systems produced the same results over every simulated day.

Optimization agent
Simulation model development follows five stages: systems analysis, where key elements, interactions and behaviors are determined; system synthesis, where a complete model of the interactions and elements of the system is produced along with supporting data; verification, where the algorithmic correctness of the model is determined; validation, where model responses are compared with those of the real system that is being modeled; and model analysis or inference, where the response of the model to a range of inputs is analyzed (Mihram, 1972). Tests that provide information on the relative strengths and weaknesses of the final model solution are an important component of model analysis. The sensitivity of the model output to uncertainty in each of the parameters (eg. survival of life stages, egg batch size) provides the scientist with greater information on which parameters are the key drivers of the model output and the extent to which parameter changes may alter the conclusions that can be drawn from the model (Eschenbach & Gimpel, 1990).

The optimization agent is responsible for modifying the simulation agent so that it matches the real biological ecosystem that it models as closely as possible; such analysis of matching thus incorporates the validation stage of simulation model development (Mihram, 1972). A key prerequisite of the optimization agent is determination of the evaluation function. This gives a goodness-of-fit measure, by determining the difference between the simulated values and the real world data that the simulation must model.

Goodness of fit is measured by a variety of criteria, which must be of a form suitable for the optimization agent, namely a numerical score which must be minimized or maximized. Thus, the degree of fit is specifically defined and is not subject to variation according to the individual assessor, which can occur when simulated and observed data are compared graphically. Goodness of fit criteria will vary according to the data and the aims of the modeler. Examples include testing fit between observed and simulated values by partitioning the mean squared error based on regression analysis (e.g. used by (Rice & Cochran, 1984)) or measuring the extent to which simulated values fall within the 95% confidence intervals of the mean observed value (e.g. (Andrade-Piedra, Forbes, Shtienberg, Grünwald, Chacón, Taipe, Hijmans, & Fry, 2005)). The criteria can be applied to different aspects of the model, such as the number of generations, the timing of them and their size (Nahrung et al., 2008). Additional characteristics may also be investigated, such as model robustness in predicting the number of broods and across climate zones; further, the quality of fit across each stage in the life cycle is noted, with some being superior to others (Nahrung et al., 2008). Expressing this mathematically is challenging, since each of these aspects must be assigned a formula together with a weighting, to allow the relative importance of each aspect to be controlled and ensure the measure is not unduly dominated by some components. For example, a very simple measure might sum the absolute difference in population size between the simulation and real biological data at each day, and add to this the difference between peak timing. The total number of days for the simulation in this study is over 200, an order of magnitude larger than the number of population peaks reported in (Nahrung et al., 2008); weighting is thus required to prevent the population size measure dwarfing the peak timing, and the resulting solution prioritizing population size over peak timing.

It may also be important to measure the quality separately for each life stage, and combine these to produce the overall quality; otherwise, solutions may be produced which appear to fit well for the entire population but perform poorly for some life stages. However, a drawback of this approach is that it requires determination of preference before the optimization process is performed (Branke, Deb, Miettinen, & Slowinski, 2008); for example, the scientists need to determine whether each life stage should be weighted equally in the quality measure, or alternatively greater weighting be given to some stages. A superior approach may be to use multi-objective optimization, where each component of the quality function, such as quality at each stage, is measured separately (Branke et al., 2008). Rather than a single optimal solution being produced, such an approach results in a set of 'Pareto optimal' solutions, representing good solutions from which users may select the best (Branke et al., 2008). A solution is Pareto optimal if there exists no other solution which is superior in at least one quality measure and not inferior in any other; it is weakly Pareto optimal if there exists no other solution which is superior across all quality measures (Branke et al., 2008). The approach does not require the science subsystem members to determine their preferences for different components of the solution quality measure in advance; further, generating a collection of solutions from which members of the science subsystem can select, rather than generating a single best solution, may potentially result in better understanding of the problem (Branke et al., 2008). However, for the biological population modeling problem examined within this study the whole population is of interest. All life stages are important in determining the size of the next stage, and thus separate weightings for life stages are not required, rendering the multi objective optimization approach potentially superfluous.

A further challenge is determining how generally the model can be applied. In this study, as in most cases, there are little available data to test against; thus, any conclusions regarding model validity can only be applied to conditions within the bounds of the tested data. Where more datasets are available and can be tested, the model may be applied across a wider range of conditions.

Constraints must be used, either in terms of limiting the options available to the optimization agent or introducing penalties within the quality measure, to ensure the simulations make sense scientifically. For example, representing mortality as a negative growth rate would not make sense scientifically and yet may potentially yield accurate simulations (Debuse & Miah, 2011). Growth rate could thus be constrained to be positive, or a penalty introduced into the quality function to ensure such rates always yield solutions with poor quality measures.

The numbers used to represent the simulation parameters present another set of constraints. The Eva2 optimization framework, which we plan to use to perform the optimization within this study, supports the implementation of problems with real number parameters (Kronfeld et al., 2010). However, even if each of these has their range constrained to be one of only 100 different integer values and only eight simulation parameters are optimized, the total number of possible solutions will be 1008 (Debuse & Miah, 2011). Thus, the range of values to which each simulation parameter may be set, along with the precision with which they are represented, must be carefully set in consultation with the scientist subsystem, so that the number of potential solutions does not become unnecessarily large. Discretization algorithms have shown promise in reducing this number of solutions, although they must be used with caution to avoid problems such as increasing the problem size (Debuse & Rayward-Smith, 1999). Further, the simulation model contains a total of 55 parameters, and so even if discretization yielded only two values for each parameter there would be 255 possible parameter values if all values were optimized. Thus, if each simulation run took only 0.1 seconds then a brute force search would be infeasible, requiring over 100 million years to complete.

Unlike the simulation agent, which possessed digital ecosystem entity characteristics such as precise semantic description and a comprehensive ontology (Boley & Chang, 2007), the solution quality measure and representation scheme have no such precision or ontological knowledge base. From a social ecosystems perspective this requires considerable interaction between the technological solution development research and scientist subsystems, to determine the most appropriate measure to use. This has similarities with an ontology based decision system (Miah, Kerr, Gammack, & Cowan, 2008), which included an interaction of a technological solution with scientist subsystems, in which an interplaying role between end users and the scientific subsystems was defined. In the model, the scientific subsystems in terms of a defined problem ontology were responsible for analytics rules creation for optimized practices and assistance with improving decision support within the rural industry context.

4. DISCUSSIONS AND CONCLUSION

Through investigating biological, digital and social ecosystem interactions, this study has identified a number of practical implications and potential weaknesses in digital ecosystem agent ontologies and semantic descriptions. It is somewhat surprising that, even with precise semantic description and a comprehensive ontology, it is possible for a digital ecosystems agent such as a DYMEX simulation to prove challenging to emulate. This study suggests that these two characteristics alone may prove insufficient for an agent to be described in enough detail to be used within a digital ecosystem; a number of further characteristics may be necessary.

Firstly, a "white box" in addition to "black box" semantic description must be provided; precise details of how each module within the agent processes data must be provided, rather than simply describing the process as a whole.

Secondly, detailed "white box" descriptions for each module within an agent may result in an overall description which is too complex for human comprehension. This is particularly evident in this study, where modules process multiple cohorts over series of simulated days. Thus, some form of "debugging" may be required, whereby the agent can be executed and queried step by step and in detail, in much the same way as debuggers allow software to be executed line by line, variable values queried and so on.

Finally, even a comprehensive ontology is of little use without a mathematical ontology, describing details such as: precision level of calculations; differences between precision used within calculations and precision of reported data; and whether rounding or truncation are used, and at which points in calculations they are applied. As this study has illustrated, small errors can rapidly become significant, and thus must be eliminated as much as possible. Further, in complex models the mathematics may change from one area to the next; for example, as our results have shown, rounding may occur only in one part of an agent. This is supported by existing research which argues that languages for web ontology specification do not have sufficient expressive power to represent mathematical expressions explicitly (Annamalai & Sterling, 2003); indeed, mathematical ontologies have been developed which may provide some support in this area (Annamalai & Sterling, 2003; Davenport & Kohlhase, 2009; Gruber & Olsen, 1994).

The challenges involved in creating the second (optimization) agent do however show support for the importance of precise semantic descriptions and comprehensive ontologies (Boley & Chang, 2007), even without the additional characteristics described above. The presence of such properties would have significantly simplified this step of the study.

Collaboration between digital and biological ecosystems involves similar challenges to collaboration between fully digital systems. Digital collaborations can require alignment, mapping and translation between system rules and ontologies where these are not fully compatible (Boley & Chang, 2007). The first (simulation) agent within this study was designed to represent the rules and ontologies of a real biological ecosystem as closely as possible. The second (optimization) agent had the goal of ensuring that this alignment, mapping and translation occurred correctly, by modifying the simulation parameters to improve how it modeled the biological system. However, determining how to measure the closeness of this alignment, mapping and translation is a significant challenge; no single approach exists, with each having benefits and drawbacks, yet the approach used will significantly impact how closely the simulation matches reality. Further, the scheme used to represent the first agent within the optimization process of the second, itself a digital collaboration, has the potential to significantly affect the collaboration between the digital simulation agent and its biological counterpart. Future research will therefore explore these issues as part of achieving the key goal of ensuring that the digital ecosystems maps to its biological counterpart as accurately as possible.

The key contributions of this study thus include: investigation of practical implications of the application of digital ecosystems principles to complex interactions between biological, digital and social ecosystems; identification of potential weaknesses in semantic descriptions and ontologies of digital ecosystem agents; and the proposal of extensions to digital ecosystems to address the identified issues. The next step will be to extend the study to examine the further application of digital ecosystems principles, as the optimization agent is developed and comprehensively tested using real world field data.

5. REFERENCES

Andrade-Piedra, J. L., Forbes, G. A., Shtienberg, D., Grünwald, N. J., Chacón, M. G., Taipe, M. V., Hijmans, R. J., & Fry, W. E. (2005). Qualification of a plant disease simulation model: Performance of the LATEBLIGHT model across a broad range of environments. *Phytopathology*, 95(12), 1412-1422.

Annamalai, M., & Sterling, L. (2003). Dealing with Mathematical Relations in Web-Ontologies. *Paper presented at the Second International Conference on Autonomous*

Agents and Multi-Agent Systems, Melbourne, Australia.

Boley, H., & Chang, E. (2007). Digital ecosystems: Principles and semantics. *Paper presented at the IEEE International Conference on Digital Ecosystems and Technologies*, Cairns, Australia.

Branke, J., Deb, K., Miettinen, K., & Slowinski, R. (2008). Preface. In J. Branke, K. Deb, K. Miettinen & R. Slowinski (Eds.), *Multiobjective optimization* (pp. V-XIII). Berlin: Springer-Verlag.

Chen, Y. J. (2010). Development of a method for ontology-based empirical knowledge representation and reasoning. *Decision Support Systems*, 50(1), 1-20.

Davenport, J., & Kohlhase, M. (2009). Unifying Math Ontologies: A tale of two standards. *Paper presented at the Intelligent Computer Mathematics*, Berlin, Germany.

Debuse, J., & Miah, S. J. (2011). Optimization based Simulation Model Development: Solving Robustness Issues. *Paper presented at the 5th IEEE International Conference on Digital Ecosystems and Technologies*, Daejeon, Korea.

Debuse, J., & Rayward-Smith, V. (1999). Discretisation of Continuous Commercial Database Features for a Simulated Annealing Data Mining Algorithm. *J. Applied Intelligence*, 11, 285-295.

Eschenbach, T. G., & Gimpel, R. J. (1990). Stochastic sensitivity analysis. *The Engineering Economist*, 35(4), 305-321.

Gruber, T. R., & Olsen, G. R. (1994). An ontology for engineering mathematics. *Paper presented at the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, Bonn, Germany.

Kronfeld, M., Planatscher, H., & Zell, A. (2010). The EvA2 Optimization Framework. *Learning and Intelligent Optimization*, 6073, 247-250.

Maywald, G. F., Bottomley, W., & Sutherst, R. W. (2007). DYMEX Model Simulator Version 3 User's Guide. http://www.hearne.com.au/attachments/DymexModelSimulator3.pdf

Miah, S. J., Kerr, D., Gammack, J., & Cowan, R. T. (2008). A Generic Design Environment for the Rural Industry Knowledge Acquisition. *Knowledge Based Systems Journal*, 21, 892-899.

Mihram, G. A. (1972). Some practical aspects of the verification and validation of simulation models. *Operational Research Quarterly*, 23(1), 17-29.

Nahrung, H. F., Schutze, M. K., Clarke, A. R., Duffy, M. P., Dunlop, E. A., & Lawson, S. A. (2008). Thermal requirements, field mortality and population phenology modelling of Paropsis atomaria Olivier, an emergent pest in subtropical hardwood plantations. *Forest Ecology and Management*, 255, 3515-3523.

Rice, J. A., & Cochran, P. A. (1984). Independent evaluation of a bioenergetics model for largemouth bass. *Ecology*, 65(3), 732-739.

Soo, V. W., Lin, S. Y., Yang, S. Y., Lin, S. N., & Cheng, S. L. (2006). A cooperative multi-agent platform for invention based on patent document analysis and ontology. *Expert Systems with Applications*, 31(4), 766-775.

Yaeger, L. S. (1994). Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life in a New Context. *Paper presented at the Artificial Life III*.

Yaeger, L. S., Sporns, O., Williams, S., Shuai, X., & Dougherty, S. (2010). Evolutionary Selection of Network Structure and Function. *Paper presented at the Alife XII*.

Zhang, G., Gao, Y., & Lu, J. (2011). Competitive Strategic Bidding Optimization in Electricity Markets using Bi-level Programming and Swarm Technique. IEEE Transactions on Industrial Electronics, 58(6), 2138-2146.Wang, G., Yang, S., & Han, Y. (2009). Mashroom: End-User Mashup Programming Using Nested Tables. *Paper presented at the International World Wide Web Conference*, Madrid, Spain.

Wang, X. D., Yang, X., & Allan, R. (2006). Top Ten Questions To Design A Successful Grid Portal. *Paper presented at the Second International Conference on Semantics, Knowledge and Grid (SKG 06)*.

Xu, H., Sharma, S. K., & Hackney, R. (2005). Web services innovation research: Towards a dual-core model. *Journal of Information Management, 25*, 321–334.

Yahoo Pipes (2009). Retrieved July, 2009, from http://pipes.yahoo.com/pipes/

Yee, R. (2006). Mashups, IST-Data Services. Retrieved December, 2009, from http://dret.net/lectures/services-fall06/Mashups.pdf