Applying Complexity Theory in Business Information Systems

Dr. Albert van Aardt

Faculty of Business
Northland Polytechnic
Whangarei, New Zealand
Avanaardt@northland.ac.nz

"If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is" - John Louis von Neumann.

ABSTRACT

Complexity theory has been applied to fields as diverse as biology, economics and stock exchange trading, with surprisingly productive results. This paper investigates the implications of complexity theory in the design, implementation and maintenance of business information systems. It suggests that the implications are far-reaching and that some very fundamental changes in perspectives are called for.

KEYWORDS

Complexity Theory, Business Information Systems.

1. INTRODUCTION

1.1 Background

Many people in the industrialized world may have experienced a computer systems failure. No matter how careful systems are designed or how intensively they are tested, failures are seemingly becoming more common. Why do systems fail? What new and innovative ways can be developed to design computer systems to reduce these failures?

This paper attempts to explain some of the reasons for systems failure using complexity theory. From this some pointers will be given to a different approach to systems analysis and design.

Firstly, a brief description of complexity theory will be given, in particular how this has lead to the idea of Complex Adaptive Systems (CAS). Secondly, Business Information Systems (BIS) will be briefly described. Third, an attempt will be made to interpret BIS as CAS, and lastly, from this attempt, some tentative implications will be drawn.

The paper is speculative and its main objectives are to stimulate debate and to identify areas of further research.

1.2 Research to Date

Chaos theory, the forerunner of complexity theory, was developed by a number of mathematicians working on non-linear mathematics. People such as Lorenz, Mandelbrot, Feigenbaum, Stein, Packard, Crutchfield, Farmer, Ford and a host of others worked on such ideas during the 1960's and 70's, and found to their amazement that it is actually very difficult, if not impossible, to find "real" chaos (Gleick, 1994: 316). Indeed, when analyzed, chaotic systems merely turned out to be highly complex systems that only appear to be chaotic. To put it in another way: simple systems can often produce highly complicated results. This lead to theorizing complex systems as Complex Adaptive Systems (CAS).

One person who has probably had the most impact on the science of complex systems was John Holland of the Santa Fe Institute in New Mexico. Holland applied complexity theory to the economy, and showed that social systems (such as the economy or politics, and biological systems such as cell reproduction or ant nest) growth follow a few simple rules, yet end up becoming highly complex. (Waldrop, 1993, p. 147)

Little research on CAS from a BIS perspective has been done. Of note was McBride (1999), who pointed out that information systems function in a chaotic world. McBride poses the question as to how systems should be designed to cope with the complexities of modern business. Few authors have investigated the implications of CAS from a managerial perspective, and Phelan (1995) concludes that strategic planning is useful in a complex environment.

However, a preliminary search on the Internet has not yielded any research on the implications of CAS from a BIS perspective. Hence this attempt to address these two questions.

Can a Business Information System be interpreted as a Complex Adaptive System? If so, what are the implications for the design of Information Systems?

2. COMPLEX ADAPTIVE SYSTEMS (CAS)

According to Dooley (1996) a CAS behaves/evolves according to three key principles (quoted verbatim):

"a. Order is emergent as opposed to predetermined

b. A system's history is irreversible

c. A system's future is often unpredictable."

The basic building blocks of the CAS are agents. Agents are semi-autonomous units that seek to maximize some measure of goodness by evolving over time. Agents scan their environment and develop schema representing interpretive and action rules. These schema are by definition incomplete and changing."

From this it is not difficult to see why small changes can have large effects in a particular system. Indeed, the concept of "positive feedback" (the effect each agent has on participating in the system) has lead a number of authors to conclude that "the whole is greater than the sum of its parts" (Waldrop, 1993 p.203).

Such generalization has proved controversial. It flies in the face of conventional wisdom - such as the Law of Diminishing Returns in Economics which states the opposite (Vanderweshuizen *et al*, 1991, p. 121 to 129). There seems to be ample mathematical proof, however, that complex systems indeed exist, and collectively have greater effect than its parts can achieve. (Waldrop, 1993, p. 269 to 274)

Nonetheless, Dooley's third principle, that a "system's future is often unpredictable" is of great interest, despite the claim that these systems follow "a few simple rules". A CAS is deterministic, in the sense that the outcomes are often no more than the result of the interaction between the various agents. But a CAS is also unpredictable, because the various internal and external interactions can produce billions of different possible outcomes, of which only one actually happens. (Percival in Hall, 1992 p. 15)

3. BUSINESS INFORMATION SYSTEMS

The way in which business information systems have been conceived has changed in recent decades. In 1948 Claude Shannon proposed that a communication system consists of a sender, a channel and a receiver (see Figure 1). According to Shannon, as the signal travels from the sender to the receiver, noise is introduced along the channel that interferes with the signal. This noise has been regarded as something "bad" that must be filtered out or somehow discarded.

According to Laudon and Laudon (1996, p. 9), an information system consists of a number of "interrelated components working together to collect, process, store and disseminate information". Stair, (1992: p. 17) stated that these components are computer hardware, software, telecommunications, people and procedures. A business information system (BIS) is therefore an information system used in a commercial environment. The classical concept of information systems are illustrated in Figure 2.

This input-processing-output (IPO) model of information systems has been the dominant view since the late 1950's. Initially, programs were written to manipulate data in large sequential files, but this soon was found to be too limiting. In 1970 Codd proposed that there are functional relationships between data items, and that it is important that these relationships be used to produce information systems that are more attuned to the real world. This lead to a whole new paradigm in business systems. Relational theory, as it came to be known, was an attempt to overcome the problem that the "hard coding" of data structures presented, namely that circumstances change rapidly, and that information systems had to be able to reflect these changes.

Towards the end of the 20th century, Object Orientation (OO) was in an attempt to represent a better view of the real world situations. OO uses the so-called "Use-case" scenarios to purportedly reflect actual, real life situations (Lausen & Vossen, 1998, p. 159-172). OO systems are considered by some to be the best approach to find solutions in a changing operational environment.

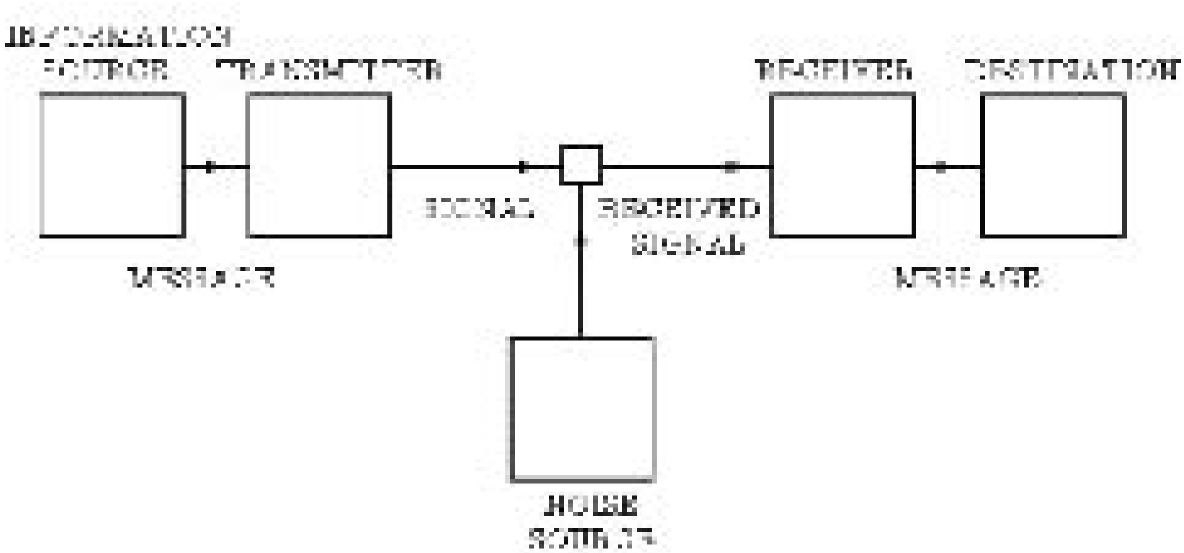

Figure 1

Shannon's Model of Communication

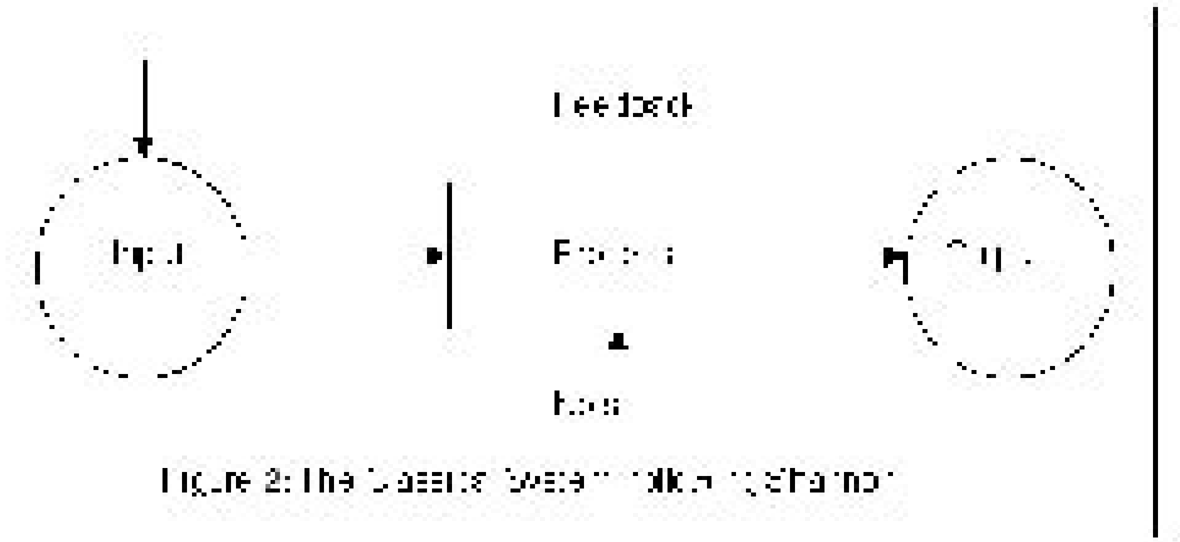

Figure 2
The Classical View of Information Systems

4. INTERPRETING A BIS AS A CAS

4.1 Humans and Business Procedures as "agents"

Is the "semi-autonomous agent" as a concept in a CAS theory the same as a "user" in a BIS? A "user" can be seen as an entity which acts semi-autonomously, since this person has reasons for interacting with the system on their own changing terms. For example, a customer at an ATM may wish to withdraw money; another customer may wish to deposit money, a third may wish to do both and then change their mind. Similarly, the "internal" users of a BIS can be seen as "agents" - the bank clerk wanting to update a customer's account, for example. Indeed, in OO parlance these "users" are referred to as "actors". Further, "actors" can also be other systems. One system can deliver output which another system then takes as input for further processing. An example would be my personal computer system interacting via the Internet with another computer system, to cope with my changing demands, all in a constantly changing information environment.

According to complexity theory, however, the results of a CAS are unpredictable. If a BIS is indeed a CAS, then the results (outcome) from a BIS must also be unpredictable. And since BIS fail because they are highly complex, and because of the positive feedback loops within such a system in themselves create changes that may influence the outcome, systems failure is likely, especially if demands exceed design specifications.

Furthermore the business environment in which the BIS must operate is one characterized by rapid changes. If the economy is indeed another CAS (as Holland pointed out) and if the BIS must operate within this environment, then a BIS may be interpreted as a type of CAS, provided the systems designer is another active agent.

The interesting point here is that humans, in interacting with one another and with a BIS, often tend to get involved in a positive feedback loop, using output from the system to generate input again. It is not uncommon to find people using a BIS for something it was never designed to do. An example is a spreadsheet package is used to build a sophisticated accounting system. Though not necessarily the best way to use the software, such adaptations happen frequently. The proliferation of systems being developed using relatively simple tools such as Visual Basic and MS Access is another example. Generalized software packages seem to foster an almost organic development of business-specific applications .This phenomenon illustrates complexity theory: "Order is emergent as opposed to predetermined."

4.2 Hardware, Software and Telecommunications as "agents"

To the other components of a BIS: are not so easy to define as "semi-autonomous agents". Computer hardware cannot "adapt" by itself. Any changes to the hardware need the intervention of an outside agent, a designer. If the designer and the supportive social and business environment are considered components of the BIS, it meets the definition of a CAS.

With regard to software stronger case can be made. Although it is true to say that, in commercial systems, once a program has been written, it cannot change by itself, it is equally true that programs can be written in such a way as to be adaptable according to the circumstances.

For example, programs are often written in such a way as to be driven by parameters. Given one set of parameters, the program would process the data in a particular way. A different set of parameters would invoke a very different processing cycle. The concept of a "schema" as used in complexity theory is highly analogous to the algorithm of a computer program.

Recent design approaches, such as the so-called "auto-repair" facility in MSOffice 2000, is another example of software with a certain degree of "autonomy". Error correcting software, as used widely in telecommunications, have a certain degree of flexibility built in that allows some adjustments to be made depending on the real life environment. Similarly, the routing of connections via the Internet is handled in such a way that the actual routes are dynamically adjusted to suit the circumstances.

Other developments in the software industry may also be interpreted in this manner, i.e. software that is capable of changing to suit the environment. This capability can be observed more clearly in the OO paradigm, where each object reacts to other objects. A certain amount of "freedom of movement" can be observed. Not that objects can be defined as semi-autonomous agents. On the other hand, "intelligent agents" and "web bots" may be seen as programs with a fairly high degree of autonomy. When used, for example, for data mining applications, intelligent agents can produce some startling and rather "clever" results. In a similar vein, search engines on the World Wide Web can deliver some very useful information (e.g. www.google.com).

Newer developments of software, such as artificial intelligence (Turban, 1995, pp. 442-471), neural networks (pp. 682-756) and even expert systems (Mallach, 1994, pp. 442-477) can all be interpreted as attempts to give computer programs the ability to change themselves - i.e. to become closer to the concept of "schemas". (Although, as a student of mine recently pointed out: "Artificial intelligence is still no match for natural stupidity!")

4.3 "The System's History is Irreversible"

Business data often depend on a sequential order of events to have any meaning. For example, before money can be deposited into a bank account, the account must first be opened. An account can only be closed if it has previously been opened. Data mining depends heavily on the assumption that transactions occur in a sequential manner. This sequential mindset has had a great influence in the early system design to the point that monolithic systems were designed to be exclusively sequential.

More modern systems no longer use a purely sequential approach, although "event-driven" programming can still be seen as a sequential approach to a degree - one thing happens after another. This sort of sequential thinking can also explain why many business systems are designed and implemented in a linear fashion.

Clearly, a BIS must cater for the sequential flow of events in the real world. However, what is often overlooked is the interaction between the various actors - for example, a person receiving a salary (i.e. a deposit transaction in their bank account) also means that the employer had to have an equal withdrawal transaction - in other words, the real world is of a non-linear nature. It is simultaneous, and is subjected to retrospective reinterpretation by the actors.

Concepts such as database roll-back and roll-forward techniques can be seen as attempts to maintain the logical, sequential nature of transactions - i.e., an attempt to maintain the integrity of the database; or, to put it in terms of complexity theory: to maintain the empirical or factual history of the system.

5. IMPLICATIONS AND TENTATIVE CONCLUSION

A number of fairly recent initiatives in the software industry can be interpreted as being attempts to make software more like "autonomous agents". It would also appear that, instead of becoming more robust, information systems are actually becoming more prone to error than a few years ago. According to complexity theory, the more agents that are involved in a system, the higher the probability of "unpredictable results" such as a systems failure. Indeed, when one considers the millions of transistors, instructions, transferred bits, possible logic branches and all of the ancillary equipment that all have to perform with absolute perfection, it is a wonder that computer systems work at all.

Given the inevitability of unpredictable results, the concept of a BIS should not use linear modeling based on Shannon's view, but use a more realistic approach.

Since it is impossible to predict exactly what the final outcome of a CAS will be, it might be wiser to assume that the actual outcome would probably not be the same as the designed output. Indeed, anecdotal evidence suggest that very rarely does the planned output and the actual outcome of a BIS match. Furthermore, expectations of users are often much higher than what can initially be delivered (Duffy, 1993). As systems get more and more complex, complexity theory indicates that not only is it impossible to predict the final outcome, but relatively

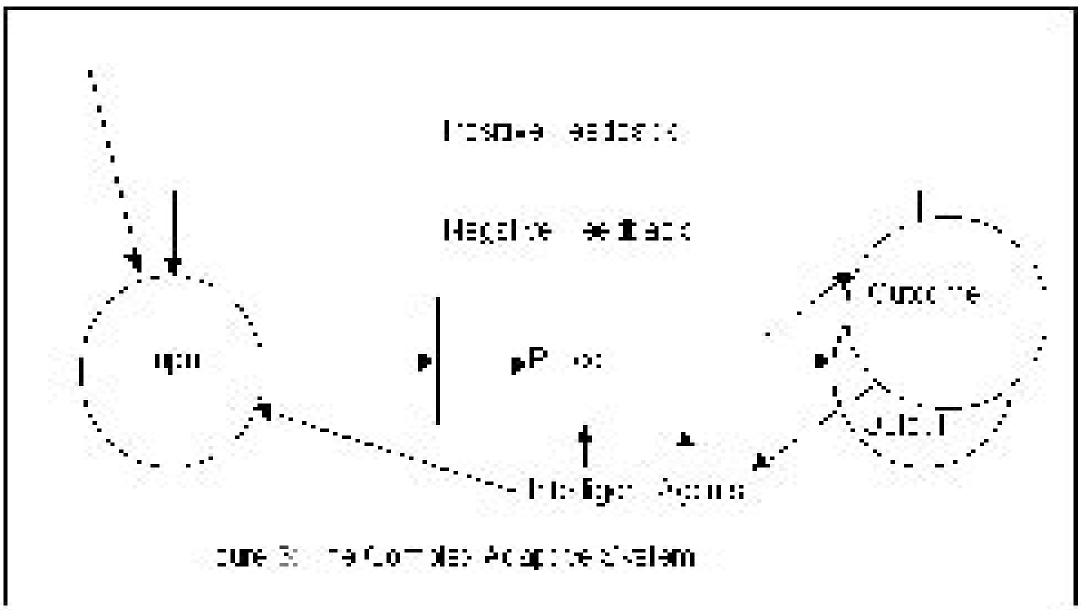

Figure 3

A Complex Adaptive System

small inputs can have major effects. Equally true, of course, is the fact that minor inputs often have no discernable impact on the final outcome. The point here is the unpredictability.

Information systems design has tried to cope with the unpredictable nature of the real world. Techniques such as relational theory, OO, event-driven programming, artificial intelligence and self-repairing software can all be seen as attempts to cope with uncertainty. However, if BIS had been considered to be CAS from the start, design approaches may very well be quite different.

Although the sequential (historical) flow of transactions are important, systems should not be designed from solely this aspect. Systems should rather be designed from the perspective of the user (or client). In other words, it is not the actual transactions themselves that should be focussed on, but rather how the user acts in the real world, and the BIS should reflect this. The object “Joe Soap” is an instance of the class “Customer”, but he could also be an employee, a parent, a motorist, a tourist, a patient, etc . All of these roles are important at some point in time, and all of these roles actually interact with all the other roles, not to mention with other “objects”. The OO paradigm cannot solve the problem of a dynamic environment; objects do not (as yet) act semi-autonomously.

The approach to systems design widely used appears to have been limited by the competing claims of “data centric” and “process centric” (see figure 4). A non-linear view should be adopted. Complexity theory, as manifested in CAS, appears to be a more realistic view of the real world. Using this approach, it would mean some radical changes in both systems design as well as in programming technology. Programs need the to be self-adapting. Some advances have been made in neural networks in this regard; but it will be some time before commercial systems can be developed easily. Some OO software vendors, such as JADE, seem to be developing a useful programming environment, although there is still a lot of work to be done before the non-linear characteristics of the real world can be fully accommodated.

Positive feedback and “noise” might need to be taken seriously, rather than discarded or filtered out. Comments from users on aspects of a system may be very useful. Users do not want to be held back by the system; they simply want to get on with the job. For the clerk, the job is to issue an invoice (for example), and not to go through a tedious process of getting the system working. The system should be “transparent”, and the users should only “see” the job they are trying to complete, not anything else. The term “user friendly” has been abused to the point of extinction, but the principle remains the same: systems should be there to do a job, and users should not be frustrated in achieving their objectives. It follows that feedback and “noise” should be accommodated in an ongoing re-design process.

If a BIS is in fact a CAS, it would imply that the best approach to systems design would be small, incremental steps, rather than via grand one-off design. OO takes this approach and develops relatively small parts of the system. Implementation and design can happen almost at the same time, and positive feedback can be incorporated to a degree. However, as stated earlier, in order to have the system to act in a self-adjusting way, we need new software languages - and probably even new operating systems and hardware architectures. At the same time, the sequential nature of business processes will need to be embedded in a historical context.

BIS’s in use fail frequently because the complexities of the system tend to be ignored both during the design phase as well as during the implementation phase. It would make sense to approach systems design with the principles of complexity theory firmly in mind rather than setting them aside. The field of IS lacks the tools to produce fully commercial systems that are self adjusting, but if it can take cognizance of complexity theory in current designs, and reduce the number of failures. For instance, assuming that the system will fail sooner or later would lead to more sophisticated disaster recovery and data replication strategies. Conversely, understanding that complex systems are more prone to failure than systems of a more simple design will also influence our thinking.

Information systems as used in the business environment have become ‘mission critical’; the modern business can no longer function without a good information system. It would seem reasonable to approach systems design from the same

Process orientated	C, Fortran, process control	JAVA, C++, VBasic, UML
Data orientated	3GL, COBOL, periodic batch systems, relational DB	SQL
	Sequential, transactional, batch view	Event driven, on-line, user involved view

Figure 4: Approaches to systems design
(Source: own design)

perspective as that in which the business operates, i.e. complexity. Much more research is needed in this field.

BIBLIOGRAPHY

Codd, E.F. (1970) "A relational model for large shared data banks" CACM 13:6 June 1970 pp. 377-387

Dooley, K. (1997): "A Complex Adaptive Systems Model of Organization Change," Nonlinear Dynamics, Psychology, & Life Science, Vol. 1, No. 1, p. 69-97

Duffy, N. (1993) "User expectations and technological deliverables". Presentation delivered at a meeting of WG4 (Electronic Commerce) SABS, Pretoria.

Gleick, J (1994) "Chaos" Abacus, London ISBN 0 349 10525 1

Hall, N.(ed.) (1992) "The New Scientist guide to Chaos" Penguin London

Holland, J.H. (1995): "Hidden Order": Addison-Wesley. Reading, MA

Laudon, KC & Laudon, JP (1999): "Management Information Systems" Prentice-Hall,

Lausen, G. & Vossen, G, (1998) "Models and Languages of Object-Orientated Databases" Addison-Wesley, Reading Mass. ISBN 0-201-62431-1

Mallach, E. G. (1994) "Understanding Decision Support Systems and Expert Systems" Irwin, Burr Ridge, Ill. ISBN 0 256 11896 5

McBride, N (1999) "Chaos Theory and Information Systems", Department of Information Systems, Faculty of Computing Sciences and Engineering, De Montfort University, Leicester

Phelan, S. (1995) "From Chaos To Complexity In Strategic Planning" Paper presented at the 55th Annual Meeting of the Academy of Management, Vancouver, British Columbia, Canada August 6-9, 1995

Shannon, C (1948) "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. 27 (July and October 1948), pp. 379-423 and 623-656

Stair, R. (1992) "Principles of Information Systems". Boyd & Fraser, Boston. ISBN 0 87835 789 0

Turban, E. (1995) "Decision Support Systems and

Expert Systems" Prentice-Hall, Englewood Cliffs, NJ. ISBN 0 13 320383 2
Van Der Westhuizen, A.I., Van Zyl, G, Botha, R.F. and Mulder, W.M. (1991) "Mikro-ekonomie" Decorata Publishers Johannesburg.
Waldrop, M.M. (1993) "Complexity", Penguin Books, London. ISBN 0 670 85045 4

Some interesting websites.

http://www.eas.asu.edu/~kdooley/casopdef.html
http://www.zweb.com/apnonlin/
http://jonescenter.wharton.upenn.edu/Default.htm
http://www.wfu.edu/~petrejh4/chaos.htm
http://www.wfu.edu/~petrejh4/chaosind.htm
http://netra.exploratorium.edu/complexity/CompLexicon/chaos.html
http://lslsun.epfl.ch/~moshes/caslinks.html
http://www.jeffsutherland.org/oopsla98/rostal.html
http://www.cs.iastate.edu/~honavar/alife.isu.html
http://www.casresearch.com/
http://www.workfrontiers.com/CAS.htm
http://www.cea.uba.ar/aschu/complex.html
http://www.haskins.yale.edu/Haskins/MISC/DEST/complexity.html
http://www.emergence.org/Complexity1.htm
http://www.brint.com/Systems.htm
http://www.csu.edu.au/ci/
http://complex.csu.edu.au/complex/
http://www.human-nature.com/science-as-culture/rosenhead.html
http://www.mcu.edu.tw/department/pubaffair/parn/pais/org/org-complex.htm
http://www.brint.com/Systems.htm
http://www.cs.buffalo.edu/~goetz/al.html
http://www.cudenver.edu/~mryder/itc_data/complexity.html
http://serendip.brynmawr.edu/complexity/newton/
http://www.aic.nrl.navy.mil/~spears/generators.html
http://www.santafe.edu/sfi/publications/Bulletins/bulletin-spr97/TEXTW1.HTML
http://www.iiasa.ac.at/
http://www.santafe.edu/sfi/publications/Bulletins/bulletin-spr95/10control.html
http://www.primenet.com/~terry/index.html
http://www.calresco.org/links.htm#frac
http://www.brint.com/papers/change/
http://www.santafe.edu/
http://www.cio.com/archive/enterprise/041598_qanda_content.html
http://www.econ.iastate.edu/tesfatsi/sylalife.htm
http://www.calresco.org/
http://inls.ucsd.edu/
http://serendip.brynmawr.edu/complexity/sierpinski.html
http://academic.brooklyn.cuny.edu/education/jlemke/theories.htm#Self-organizing
http://www.aic.nrl.navy.mil/~spears/theory.html
http://agents.umbc.edu/
http://www.stern.nyu.edu/informs/complexity.html
http://www.cg.tuwien.ac.at/research/vis/dynsys/index.html
http://www.agentbase.com/survey.html