A MODEL OF AN ABSTRACT INTELLIGENT ENTITY AS A MEANS OF COMPARISON OF INTELLIGENT DATA OBJECTS (IDO) AND THEIR MANAGEMENT SYSTEM (IDOMS) TO BIOLOGICAL ORGANISMS AND ECOLOGICAL SYSTEMS

by

STEVEN C. HANSEN

B. S., St. John's University, 1978
M. S., Kansas State University, 1983

A MASTERS THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1987

Approved by:

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to extend my sincere thanks to my major professor, Dr. Beth Unger, for her time, support, and encouragement throughout the development of this thesis. I would also like to thank my committee members, Dr. Rich McBride and Dr. Austin Melton for their time, efforts, and suggestions. I would like to extend my special thanks and appreciation to my wife, Laura, for her patience, love, and understanding throughout the term of my graduate research.

Chapter 1

Introduction

The field of computer science has burgeoned over the past several decades, rising from nonexistence to the massive entity which it is today. During this time period the science of programming a computer has developed from requisite hardware changes on early computers to present day production of high level software. In this same time period, hardware systems have grown from small single machine architectures to colossal networks of computers and peripherals. Simultaneously, software has evolved from machine language programs with direct hardware access to complex, integrated software systems [Deit84]. Current complex integrated software systems include operating systems for single machines and small networks [Tane81]. Large conglomerations of software also exist in large networks, but well integrated operating systems for large networks do not currently exist [Tane81]. The reason for the lack is the complexity of design of such systems coupled with the high cost of transition [Tane81].

Much research has been done in recent decades on the behaviors of large scale systems. These systems can be loosely defined as any systems which must be partitioned into a number of interconnected subsystems for either conceptual or computational reasons [Silj78]. Systems which can be classified as large scale systems include ecological systems, economic systems, and engineering power systems

[Silj78]. Intuitively, one realizes that large computer networks would also fall into this category of systems, as no one model of such systems would serve to explain the interworkings of all of the components of such a system.

Large scale system research has uncovered properties that a large scale system must have to possess stability. Some of these properties relate to the amount of cooperation and competition which exists among component parts [Silj78]. In large systems, subsystems must have cooperation among members, while it is necessary that there be less cooperation between members and nonmembers of any given subsystem. If the large system is not subdivided in this manner, perturbations (removal and additions of subsystems) can cause a lack of stability in the system at large. Design of the overall system will have to balance the levels of cooperation and competition among all components of the large scale system. Since cooperation and competition can occur in many different categories of interaction, programming of such a system can become a difficult and extremely complex task.

Within the field of computer science, research in concurrent processing has exposed many previously unknown interactions among system components. Some discoveries of such interactions have been made following observation of unpredictable system behaviors. Much information on interactions which exist among components of current systems has been derived from systems operating on single machine architectures (i.e. databases, operating systems). These systems are created through careful consideration of the component processes and their known interactions. The behavior of such systems is controlled through control of the behaviors of the individual processes making up the system.

In recent years, there has been an increase in research directed toward creating concurrent systems within networks of computers. These systems benefit users by providing the facilities to share expensive and/or large hardware and software resources among a large group of users, thereby reducing individual costs. Simultaneously, distributed systems have proven to be a headache for system administrators. In this environment the number of problems related to producing correct systems has increased exponentially. Politics, hardware updates, and various logistic problems may well make the development of a large network operating system a difficult feat to achieve. Meanwhile, the interactions of processes within these networks create a system over which no group can effect total autocratic control. An information resource for the development of processes which can operate effectively in such environments is clearly needed. If a resource on the theory of large scale systems could be found which possesses many similarities in the types of interactions that occur among the components of computer networks, it could prove invaluable in time and monetary savings.

An area of concurrent research in the last decade concerns the task of developing office automation systems. Within this area, the development of electronic forms and form processing has received much attention. The resultant research has reached the point where forms can be defined which can somewhat autonomously route themselves around an office network, requesting selected information and authorizations from people, workstations, and databases at various nodes. These forms can fork to allow for parallel processing when appropriate, and need only return to the form originator when the form has been completely processed. This method of form handling has the advantage of freeing the processing of the form from the constant attention of a human operator, thus making the form completion process independent and potentially faster.

One possible method of increasing the efficiency of form processing is to encode within the form instructions or conditions which will allow the form to be dequeued and sent elsewhere when the form has been in a queue longer than some pre-specified time. This would allow the form to operate in a non-deterministic fashion and would thus allow it to respond to system conditions.

The electronic form definition described above is a result of research which is being conducted at Kansas State University. The name given to this class of form is Intelligent Data Object (IDO). The system within which an IDO operates has thus been called an Intelligent Data Object Management System (IDOMS).

Although several aspects of Intelligent Data Objects have already been developed, no implementation of a model Intelligent Data Object Management System has been built at this time. System design techniques often rely on an examination of the behaviors of similar systems which are already in existence. Design of an Intelligent Data Object Management System is thus made more difficult due to its dissimilarity with current systems. General system theory can provide ideas on overall system behavior and control, but is probably not sufficiently detailed enough to suggest efficient methods of implementing Intelligent Data Objects such that they can accomplish their goals within an unpredictable and changing network setting. Although there does not appear to be a system already existing in the computer science field which can be effectively utilized for IDOMS design principles, there may exist such a system outside of the realm of computer science. We posit that biological systems are sufficiently complex so as to make them viable candidates as systems to utilize in modeling IDOMSs.

Biological systems have been subdivided into numerous subsystems by scientists. Of these subsystems, ecological systems appear to most closely match IDOMSs. Ecological systems consist of many component parts (organisms and their non-living environment) which have no high-level common goal. Organismic systems and subsystems, on the other hand, operate toward the common goal of survival of the organism or reproduction of the organisms genes. Higher level systems (i.e. biosphere, etc.) do not share enough interactions to effectively model IDOMSs. Ecological systems therefore appear to closely model the important components of an IDOMS.

We believe the concept of IDOs operating within IDOMSs is similar to the concept of organisms operating within an ecosystem. Organisms make decisions based on the current state of their environment. These decisions are usually beneficial to the organism in that they allow the organism to continue to exist and/or reproduce. Tsichritzis [Tsic85] has already given an anecdotal, non-rigorous, account of the relationship between form-based objects and animals. His account is meant more as an interface to help end-users visualize such objects.

Ecological literature represents a large body of knowledge about the interactions among organisms and between organisms and their environments. Ecology therefore may contain information which could be used to construct efficient IDOMSs without vast duplication of effort in determining the properties of these systems.

In this thesis I have established a basis for comparison of ecological systems and IDOMSs. Upon establishing this basis, a few system principles have been extracted from ecological systems and reapplied to IDOMSs in an attempt to demonstrate the utility of this approach. Finally, suggestions are made as to additional principles which might prove to be fruitful areas for further research. Other areas in computer science which might also benefit from this approach are discussed.

Chapter Two

Intelligent Data Objects and the Management System (IDOMS)

This chapter has two sections. The first section will define Intelligent Data Objects and the Intelligent Data Object Management System (IDOMS) at their current level of development. The second section will abstract the essential qualities of an IDOMS which are applicable toward the goals of this thesis.

2.1 A description of the current state of IDOMSs

Until recently, most information processing in office environments has involved the filling out and maintenance of paper forms. These forms were often processed sequentially, the form being passed from person to person, each of whom provided the applicable information. Each form was often also "adopted" by a human who took responsibility for its proper processing. Due to storage, processing, and transportation problems associated with the processing of paper forms, there has been a recent effort to automate the process of handling a form. The process of automation involves replacing the traditional paper format with an electronic counterpart which can be processed, stored, and transported within distributed computer systems.

The automation process can be logically subdivided into two major parts: the representation of the electronic form itself, and the representation of the software environment within which the form exists. One possible model of an electronic form handling system which contains both of these parts has been developed at Kansas State University in recent years. A generalized description of that model follows.

2.1.1 The Intelligent Data Object

The development of a model for electronic forms at Kansas State is based on a framework defined in 1983 by McBride and Unger [McBr83]. They describe five major components which are necessary to model the processing of a unit of work or a job in a distributed environment (Figure 2.1).

1. a structural model for each procedure or function provided in a distributed system,

2. a structural model of the control program that directs the processing of an individual job,

3. the status of a job (including both its control and data state),

4. global information accessible to all jobs and functions in the system,

5. data files (whose usage may be restricted and therefore do not represent global information).

Figure 2.1 - Five major components necessary to model the processing of a unit of work or a job in a distributed environment [McBr83]

McBride and Unger describe one possible model of the processing of jobs in a distributed environment. Their model utilizes Petri nets to describe each of the procedures that a unit of work must invoke. A Petri Net is also used to model the control program which oversees the execution of available procedures. A "token" marks a job's location within the Petri net. Carried in the token is information vital to the job's execution. The necessary contents of a token in the McBride-Unger model are given in Figure 2.2.

1. "Local data" which can be thought of as corresponding to the fields of a form. Some of this data can be initialized to standard or default values while others can only be filled during processing.

2. A control Petri net which dictates the manner in which the local data is processed by procedures. This control net performs a function analogous to that which a control language program performs for a job.

3. The current position of the token in the control net is stored in the token. Also, this token can appear in the marking of a Petri net corresponding to a procedure which is being carried out on behalf of the token, so that its position in this net must be recorded too.

4. The token may carry a capabilities list which describes the files that a token has access to and also the operations which may be carried out on those files

5. The token may also carry with it a history of the transitions that have been executed on behalf of the token. Such a list can include the time at which a transition was first enabled, and also the time at which the transition completed its function and introduced the token into its output stream.

Figure 2.2 - The components contained by a token in the McBride-Unger Model of jobs operating in a distributed environment [McBr83]

Since publication, the model proposed by McBride and Unger has been further developed through a research project which ultimately produced several Master's Theses in the mid-1980's. The goal of the project was to give a more detailed definition of the McBride-Unger model as applied to the idea of an electronic form. The electronic form thus developed was named an Intelligent Data Object (IDO) and the software environment within which it operates was called an Intelligent Data Object Management System (IDOMS). An intelligent form has also been defined. An intelligent form is a version of an IDO which allows access by multiple users to its local data and allows concurrency to exist among its internal procedures [Lee86].

Following is a brief description of the theses and a dissertation involved in the development of the concept IDOs and IDOMSs, along with a summarization of the research results.

[Bish86] Access Rights for Intelligent Data Objects.

"The structure of information relating to the access rights of an IDO instance is defined in this paper. An implementation demonstrating the use of access rights within the IDO is included." [Bish86]

[Busa85] The Intelligent Data Object and its Data Base interface.

Provides a description of the data base interface by which the IDO is able to access some of its information.

[Gant85] Management of an Intelligent Data Object, Masters Report,

Kansas State University, Manhattan, Kansas 1985.

This report explores the components necessary in an IDOMS for the node managers and describes a prototype implementation in SHELL of such a system.

[Hone87] A Form Definition Language for an Intelligent Form.

"The formal specification of a form definition language is described. A new form based model for office automation is defined informally." [Hone87]

[Huml86] Intelligent Data Object Management System (IDOMS).

"This report describes a prototype design for an Office Information System using an Intelligent Data Object (IDO) [Huml86]."

[Lee86] Privacy and Security of an Intelligent Office Form.

"The design of a central "security manager" to protect an IOF against unauthorized alteration or destruction is presented." [Lee86] Three different security measures are used: a user profile to specify access rights, a time stamp ordering technique to specify when operations are allowed, and an encryption procedure for protection during transmission.

[Sewc86] Form Definition Language for Intelligent Data Objects.

This thesis specifies a Form Definition Language to support the implementation of an Intelligent Data Object (IDO). The Form

Definition Language provides the user interface through which the definition of an IDO can take place.

[Ryko86] Design of the IDO for the Intelligent Data Object Management System (IDOMS) Project.

The physical design of the IDO is specified.

An IDO has been loosely defined as an instance of an intelligent abstract data type [Busa85]. An intelligent abstract data type is an abstract data type possessing a self-contained set of instructions that can be routed within a potentially distributed network in a similar fashion to the routing of a message. An IDO is designed to perform some specific function, generally to function in a manner which would be consistent with the activities which generally occur in an office environment [Huml86].

McBride and Unger have listed the information required of an object which is capable of routing itself around a system based on system states (see figure 2.3).

This information is required to allow an IDO to accomplish its processing tasks within a system [Huml86]. It is not necessary that all information required by an IDO be resident within a particular IDO itself. Figure 2.4 summarizes the information which is to be contained within the IDO. A physical structure has been defined for an IDO which incorporates this information [Huml86].

A more abstract view of an IDO could liken it to an actor which

1. the data instance (a single data record)

2. processing instructions (they suggest a control Petri Net)

3. a token to identify the current position (in the Petri Net)

4. routing history

5. a capabilities list (files to be accessed with operations for those files)

Figure 2.3 - Information required of an object which is capable of routing itself around a system based on system states [McBr83]

1. unique key which identifies the form

2. the data instance

3. list of destinations

4. sender identification (originator)

5. routing history

Figure 2.4 - Information required of an IDO which resides within the IDO [Huml86]

interacts with other actors via messages [Huml86]. These actors execute asynchronously upon receipt of a message according to a presubscribed "script". As the IDO moves throughout the system, it will behave according to a predefined pattern of behavior, defined as a set of instructions at the time of form creation. The difference between an actor and an IDO is that the actor in an actor-based system derives its intelligence from a script, whereas the IDO carries its intelligence within the object itself.

2.2 Abstraction of an Intelligent Object System

As already stated, an Intelligent Data Object Management System can be subdivided into two parts, the Intelligent Data Object itself, and the distributed computer system within which it operates. The first part of this discussion will be divided between these two parts, followed by a description of the interfacing of these parts into the system as a whole.

2.2.1 Components of an Intelligent Data Object

An Intelligent Data Object must have certain components to allow it to exist and accomplish its goals. These components include an instruction set, implementation, goals, and environmental senses. Each of these components will be discussed in further detail below.

2.2.1.1 High level instruction set

An IDO needs an instruction set to accomplish the goals of its originator. These instructions should be high level instructions to make algorithm definition easier and less prone to error. A sample of possible instructions which might be represented in this set is given in figure 2.5.

1. wait
2. move in current direction of movement
3. obtain data/authorization
4. decide on next direction of movement
5. decide on next action
6. access system information
7. fork
8. join
9. return to station of origin
10. die

Figure 2.5 - High level instruction set for Intelligent Data Objects (IDOs)

Some of the instructions listed in Figure 1.5 may be too vague or general to allow their inclusion in an actual implementation of an IDO. They should give an idea, however, of the types of actions an IDO might need to take.

2.2.1.2 Implementation details of IDO which may affect processing efficiency

Choice of an implementation for an IDO can affect the processing efficiency of that IDO. Choices should therefore be made with an eye towards their effect on future efficiency. Some of the possible implementation details which can be manipulated to affect the efficiency of the final form are given below.

- physical size (bytes)

    Larger processes take longer to be routed around the network. Also, with large program size, swapping in and out may vastly increase the processing time. Finally, large processes may decrease efficiency of other programs "working" for the same company (along with their own decrease).

- size of instruction set (relative to minimal set)

  Complex instructions may result in slowing down rather than speeding up processing. There should be some optimal size of instruction set where the benefits of of the instructions present outweigh the additional IDO size they represent. Relative size is important, especially if some threshold has already been surpassed (such as the ability to be completely represented in RAM.

- quality of instruction set

  Some instructions may be more effective than others in increasing efficiency. In addition, there may be a combinatorial effect (some combinations of instructions may be more effective than others).

- ability to fork (and to what degree)

  The tendency to fork may be an important factor in determining IDO efficiency. If overhead is associated with forking then the process should only fork if forking increases average efficiency.

- priority

  The priority of an IDO can be an important factor. For example, if a low priority process is waiting in a priority queue the probability of being served in reasonable time may be small. In such a case the process may want to care for other needs first if doing so will increase its overall efficiency.

- system information

How much information should the process access how often? There is usually overhead associated with accesses to system information. Access may allow the IDO to make more "informed" decisions, thus increasing efficiency. At some point however, costs of access will outweigh the benefits.

2.2.1.3 Goals of an IDO

The goals of an IDO are important in helping to determine whether individual actions are worthwhile (see Figure 2.6). Similarly, these goals may be helpful in selecting the correct set of instructions (there may be interaction effects among instructions which increase or decrease their effectiveness).

1. to obtain a set of data and/or authorizations in a given amount of real time

2. to leave data and obtain verification of this transaction within a given amount of real time

3. to fulfill one or both of the above requirements while utilizing as little time and money as possible

Figure 2.6 - Proposed Goals for an Intelligent Data Object operating within an IDOMS

2.2.1.4 Environmental Senses of Intelligent Data Objects.

The environmental sensing capabilities of an IDO are a very important component of its implementation. In crowded, high load systems an IDO may be able to accomplish its goals through decisions based on current system states. To make these decisions it is necessary that the IDO be able to access the proper system information in some manner. A list of system states which might yield valuable information for the decision-making process of IDOs follows. An explanation of how this information might be utilized to increase the processing efficiency of an IDO is included.

System Load - Can be used to determine the probability of obtaining CPU time. A measure of the competition for CPU time within a subenvironment of a computer network.

Queue type - may provide useful information for some processes as the probability of being serviced within a reasonable amount of time. In a priority queue, for instance, a low priority process may be indefinitely denied access to the resource.

Queue length - can be used along with Average Processing Time to give a measure of the approximate time which will be required to access a certain resource.

Average Processing Time - the length of time required for a single process within a queue to obtain a desired resource. Can be measured in terms of real time or CPU time. In the latter case, real time can be estimated with the input of System Load into the equation.

Path Length - a measurement of the path to another node in a network. Most helpful if measured in real time, unfortunately this measure may be highly variable due to variable system loads. May need to be estimated from a combination of current system information and the history of the system information (e.g. is system load less at night?, etc.)

Process Priority - the priority of the process within a system. Useful in determining real time estimates in systems which use priority queues.

Optimal Path - provides information on the fastest path to a given resource within a network. Can be highly variable due to system loads, down nodes, priorities. Probably NP-hard problem to calculate in real time in large networks, good approximations may be available or calculable.

Real Time - Real time may be used to allow low priority processes to remain dormant during times of high system use. This will potentially increase the overall throughput. Low priority processes would be allowed to run on the system during times of low load, with diurnal, weekly, monthly, and annual schedules having potential importance in allowing prediction of the network/system environment. More accurate knowledge of the environment will allow the process to operate in an optimal fashion.

Unavailability of a resource - If a resource is unavailable, a process may be able to follow an alternate strategy.

Current location - Necessary for planning of movements.

Corruption of a portion of the IDO - If this information is available to the IDO, actions may be taken which will expedite goal attainment under these circumstances.

Differential system loads within a distributed system - If this information is available, better sequences of actions may be selected by the IDO.

2.2.2 Components of an Intelligent Data Object Management System.

The presence, absence, or degree of representation of the components of an IDOMS (Figure 2.7) will collectively determine the optimal behavior pattern of an IDO. An infinite number of combinations of these factors at their various levels are possible, and each will require a different set of instructions for optimal IDO behavior. A finite subset of these combinations will likely be the most common conditions of a given system, and reasonable instruction sets can therefore probably be constructed to produce instruction sets which are close to optimal for any given environment.

In this chapter a description of the abstract form of an Intelligent Data Object has been provided along with a brief review of the current level of development of the IDO. In addition, the system within which IDOs must operate was discussed, with comments made as to the possible effects of interactions between IDOs and various components of the IDOMS.

In Chapter 3 an abstract model of a biological organism and the ecological systems within which organisms operate will be defined. Chapter 4 will combine the information presented in Chapter 5 with information in Chapter 2 to produce an abstraction of an intelligent

- Intelligent Data Objects
  Other IDOs operating in the same IDOMS will affect the optimal form of the instruction set of an IDO.
- data files
- data servers
  Data servers acting as interfaces to databases can cause a bottlenecking of access to a particular group of data.
- CPU time
  single processor systems
    In single processor systems, system load may have a significant effect on IDO efficiency such that competition for CPU time will have a large effect on efficient IDO construction.
  multiprocessor systems
    Multiprocessor systems may lessen the effect of CPU competition on optimal IDO form.
- memory
  The average availability of memory for use by IDOs is a critical factor in an IDOMS. If only small allocations are available in nodes of IDO activity, an effective upper limit on IDO size may be set within a system. IDOs above this size limit may never finish their tasks within reasonable time.
- n-dimensional structure of network
  Network structure should have a large effect on optimal foraging strategies, in highly interconnected networks the number of possible alternative options grows exponentially.
- real time
  Most forms have a declining benefit function over time or at minimum have some time limit which they must not exceed if they are to be of benefit.
- humans providing data and/or authorizations
  The availability of humans for IDO service is likely to have a much higher variability than the availability of data servers, CPU time, etc.

- corrupting forces (power outages, transmission errors, humans, other processes) An unreliable system may require multiple copies of IDOs to ensure that one of the copies succeeds in completing its mission. In such a case, system performance and reliability is certain to be further degraded.

Figure 2.7 - Components of an intelligent data object management system

entity. Chapter 4 focuses on the problems and to some extent the solution of those problems in the distributed computing environment (using intelligent object) with the knowledge gained from the ecological environment.

Chapter Three

Organisms and Ecological Systems.

In this chapter we have provided definitions, in general terms, of organisms and their environment. Organisms will be defined in terms of their environmental sensing capacity, resource requirements, and basic structure. The environment is defined in terms of those components which are meaningful in relation to organisms. Basic structure is discussed first, followed in turn by the environmental sensing capacity and resource requirements.

3.1 Basic Structure of an Organism

An organism possesses several properties which can be usefully examined as a possible model for IDOs. These properties include an instruction set, environmental senses, the ability to gather and store matter and energy, and the ability to process and store information about its environment [Curt83]. This latter property is not present in all organisms, and is usually considered to be represented primarily within the higher animals. A list of these properties is given below:

- An instruction set which will allow an organism to act differentially due to varying perceptions of the environment.

- An environmental interface allowing the organism to obtain information about its environment. This interface can be varied in terms of what information is obtained and would for humans include what we classify as the five senses (but would not be limited to this information).

- The ability to store physical substances obtained from the environment.

- The ability to process and store information about the environment.

Each of these components will be described in relevant detail in subsequent sections.

3.1.2 The Instruction Set of Biological Organisms

Biological organisms carry an instruction set in the form of chromosomes contained within each cell of the organism. This instruction set is used to develop the organism into the proper functional form. These instructions can also affect behavior. Some organisms have no other mechanism of storing behavioral instructions. Others have the capacity to learn behaviors which are then stored in the central nervous system, usually in the brain. The instruction set of any given organism is usually close to that of the organism's immediate ancestors. One of the basic tenets of biology states that life begats life, in other words, no organism in the current world has been created from scratch. All organisms arise from other organisms [Curt83]. Learned behaviors can be changed greatly between parent and offspring, but few organisms have a large learned behavior component in their instruction repertoire [Klop67].

Another instruction set contained in many multicellular biological organisms is contained in the immune system. This instruction set is different from the behavioral instruction set in that each instruction operates independently, seeking out a class of polypeptides which may represent an invasion of the body by disease organisms [Curt83]. These instructions can replicate themselves if they come in contact with the polypeptide to which they are specific, thus demonstrating the ability to respond to environmental conditions. This concept is close to the concept of learning behaviors in that it allows an organism to adapt to a changing and complex environment [Curt83].

A relatively course set of possible behaviors for organisms is given in Figure 3.1. Not all organisms will possess all of the behaviors in this set, and some of these behaviors can obviously be further subdivided.

3.1.3 Environmental Senses of Biological Organisms.

Biological organisms are able to gather information about their environment through sense organs or sense organelles. A brief

1. move
2. change direction
3. feed
4. obtain environmental information
5. flee
6. fight
7. reproduce
8. decide on next action

Figure 3.1 High level set of behaviors which organisms can exhibit

description of some of the sense organs of multicellular organisms is given below along with a brief description of the types of information which these organs are capable of gathering.

3.1.3.1 Sight

Sight allows an organism to obtain information about the position and/or identity of objects in its environment. This form of environmental sensing usually requires a direct clear path between the organism and the object for which information is desired. Organs used to gather information are usually complex and are probably prone to attack by disease organisms. Loss of these organs can be disastrous for organisms depending on them as a means of gathering needed information about the environment [Curt83].

3.1.3.2 Sound Reception: Hearing

Hearing allows an organism to obtain information about the distance and/or identity of objects in its environment [Curt83]. This form of information gathering is particularly useful in environments with little or no light and in cluttered environments where direct sight paths are blocked by nearby objects as it does not require a direct path to the object to be identified. However, interference from other sounds probably causes the effective range to be greatly diminished.

3.1.3.3 Chemoreception (Taste)

Taste provides information about objects in the environment through chemoreception in the mouth of terrestrial vertebrates or over the body surface in many fish [Curt83]. Organisms which use taste are probably able to distinguish a range of properties similar to our four basic classes of taste: sweet, sour, salty, or bitter [Hick82]. The response to these tastes vary with the organism. Taste can be used in combination with smell to identify correct food items and to avoid poisonous items [Hick82]. Disadvantages of this form of environmental sensing can occur if a deadly or disabling dose of a poison is consumed before the presence of the poison is detected. The probability of consuming a disabling dose is lessened through optimal placement of the chemoreceptors in the mouth [Alco75].

3.1.3.4 Chemoreception (Smell)

Smell detects objects in the environment through chemoreception of airborne substances [Curt83]. The sense of smell in humans is relatively insensitive relative to that of other mammals, yet humans are able to discriminate some 10,000 different odors [Curt83]. One of the current theories of odor detection was developed by John Amoore of Oxford University [Curt83]. He proposed that all scents are composed of combinations of seven "primary odors". A different type of olfactory receptor would identify each of these seven primary odors. Some molecules would fit more than one type of receptor. From the response to any given molecule, the brain would thus form a "picture" of the smell allowing its classification. Smell is often used at close range, though it may be useful at considerably larger distances. Some moths can detect single molecules of pheromones and use them to track down the moth emitting the pheromones.

3.1.3.5 Touch/Physical force detection

Touch gives information on physical aspects of objects within range of physical contact by the organism [Oria69]. Information obtained includes hardness and texture. A disadvantage of touch is that it is only useful for providing information on objects within range of physical contact [Keet73,Hick74].

3.1.3.6 Heat perception

Perception of thermal energy is not usually classified as a primary sense but is nevertheless an important perceptual ability for many animals [Alco75]. Heat receptors in the skin aid many animals in identifying objects which come in contact with the skin. These same receptors are often helpful in locating an optimal environmental temperature [Alco75]. Some organisms, including pit vipers, can detect infrared radiation emitted from objects up to 8 inches away [Gamo73,Gord72]. This ability is useful for detection of warm-blooded prey items. Like touch, heat perception senses are usually only effective at close range [Oria69].

3.1.3.7 Magnetic detection

Magnetic detection senses allow organism to detect the earth's magnetic fields. Birds use perception of magnetic fields for navigation purposes [Welt82,Orr76]. Bacteria have been found which have internal magnets which cause them to swim along lines of the earth's magnetic field, thus causing them to move toward nutrient rich bottom sediments [Curt83].

3.1.3.8 Time detection

Many organisms have some sense of time. Measurement of time can occur over days, lunar months, years and other time units [Oria69]. Many cues exist in the environment which can be used to adjust behaviors to a temporal framework (i.e. position of the sun, length of day, etc.) [Curt83,Oria69]. Knowledge of time allows some organisms to budget their activities over time to take advantage of environmental conditions to which they are adapted [Curt83,Oria69].

3.1.4 Ability to gather and store matter and energy

All organisms are able to gather matter and energy from their environment and store them for future use [Curt83,Oria69]. Animals gather matter and energy through feeding while plants primarily gather matter through their roots and leaves and energy through the process of photosynthesis [Curt83]. Energy is stored in high-energy chemical compounds stored in the body of the plant or animal [Oria69]. Matter is stored similarly. Some animals are able to make auxiliary stores external to their bodies (i.e. nut and plant part storage, defense of a food resource, etc.).

3.1.5 The ability to process and store information about the environment.

Higher animals possess a central nervous system which allows them to process environmental data, thus obtaining useful information which is stored for later use [Klop67,Curt83,Oria69]. This ability enables them to live more efficiently in a changing and complex environment.

3.2 The Environment of a Biological Organism

A high level description of the the environment is sufficient to meet the goals of this thesis. Within this context, the environment of an organism may include:

1. Matter. Most discussions of matter in ecosystems are at the level of the separate elements. For the purpose of this

discussion, it is sufficient to discuss matter as having the properties of being limited or unlimited in availability over time and space. If is necessary to introduce a finer level of structure, several levels of availability can be defined between these two extremes.

2. Usable energy. Usable energy can exist in many forms: chemical, heat, light and others [Oria69,Curt83,Keet73]. For the purposes of this discussion it is sufficient to classify energy as to its availability over time and space.

3. Other organisms. Organisms which share the environment of an organism may include members of the same species, organisms which compete for matter or energy, predators, or organisms which help the organism attain its goals (often with a simultaneous benefit to the helper) [Putm84].

4. Time. All organisms operate within the time dimension. Activities must be planned to allow for obtaining resources within time constraints.

5. Space. All organisms must occupy space. Some organisms defend space due to the presence of needed resources [Smit84].

In modeling ecological system behaviors, it is useful to regroup the above environmental components into the categories of resources, organisms, time, and space. The category of resources is formed by combining energy and matter into a single category. Properties of resources exist which can be used to describe systems more accurately. The organism category can also be subdivided to make their specification more useful in the describing of system properties. Further dissection of the resource and organism categories is given below.

3.2.2 Properties of resources

The properties of resources can have large effects on the instruction sets of organisms. Resources vary in their properties, and some of this variation is quite regular and predictable over the earth's surface. Properties which hold the greatest potential for describing computer systems are described below.

3.2.2.1 Stability

The stability of a resource is important in determining its availability over time. The categories (ephemeral, perishable, moderate stability, stable) give a coarse range of descripters of stability.

3.2.2.2 Motility

Motility represents the ability of a resource to move. This term is probably only properly applied to organisms. A motile resource (organism) probably requires a different capture strategy than a non-motile resource.

3.2.2.3 Distribution of the resource over space/time

The distribution of a resource can affect the profitability of a resource by varying the cost functions of obtaining that resource. Distribution can be subclassified as given Figure 3.2.

1. spatial (distribution)
   - clumped geographically (patchy)
   - uniform distribution geographically (even)
   - random distribution
2. temporal
   - evenly accessible over time
   - clumped temporally (e.g. seasonal, diurnal changes in abundance)

Figure 3.2 - General categories of resource distribution over space and time

3.2.2.4 Abundance

Abundance is a measure of numbers of a resource that are available at a given place and time. Abundance can be roughly subdivided into the cateories of rare, uncommon, common, or abundant.

3.2.2.5 Limiting or nonlimiting

A resource is limiting to an organism if it is the resource which ultimately determines that organism's population level. The property of being limiting is usually attributed to one or at most a few resources in an organism's environment [Curt83].

3.2.3 Organisms

Organisms form a large portion of any organism's environment. These organisms can have many effects on the optimal form of the organism. Non-resource organisms can be divided into two groups, those which are motile and those that aren't motile. In addition, non-resource organisms can be divided into predators, competitors, and cooperators. Resource organisms (prey) are included in the resource section above.

3.2.4 Summary

This chapter has demonstrated that organisms and ecosystems have many aspects which are very similar to IDOs and IDOMSs. Organisms possess a structure which contains an instruction set, the ability to obtain information about the environment, the ability to store substances for later use, and the ability to interpret and store information about the environment. In addition, the environment is shown to possess qualities which makes a comparison to IDOMSs useful. In brief, the ecological environment of an organism contains resources and other organisms, all of which operate within the bounds of time and space.

In the next chapter, a model will be presented of an intelligent entity and the system within which it operates. This model will be based on a combination of the attributes of an organism operating within an ecological system as defined in this chapter and the attributes of an IDO operating within an IDOMS as presented in chapter 2. The abstraction of an intelligent entity will be defended through comparisons of organisms and IDOs and their respective systems. This model will then serve as a model within this work for organisms operating within ecological systems as well as IDOs operating within IDOMSs.

Chapter Four

An Abstract Model of an Intelligent Entity.

In this chapter a description of an abstract intelligent entity and intelligent entity based system which embodies the essential details of both organism/ecological systems and IDO/distributed computer systems is developed. This abstraction will then be utilized to justify the application of applicable biological principles to IDO/IDOMS based systems.

4.1 Components of an Abstract Intelligent Entity.

Definition: Abstract Intelligent Entity

> an entity which can adapt its method for attaining goals to the state of its internal and external environment.

The components of an abstract intelligent entity are given in figure 4.1.

4.2 Relation of the abstract intelligent entity model to existing systems

When proposing a model which serves as an intersection of the behaviors of two or more existing systems, it is important to show

The ability to take up space and possess structure.

An instruction set which allows it to react to the environment in a manner which will potentially optimize the fulfillment of its goals.

The ability to act on its instruction set.

The ability to obtain selected information about its environment.

A set of goals which include obtaining a set of resources within a given amount of time.

Figure 4.1 The components of an intelligent abstract entity

that the model applies to each of the systems involved. In this section the relationship of each component of the proposed model to both organisms and Intelligent Data Objects is discussed.

4.2.1 Structure and Space

Both organisms and IDOs possess a complex structure which they are able to maintain within their normal environment. Although the structure of organisms is far more complex than that of IDOs, the difference is much less than a comparison of either structure with the physical environment around them (excepting other organisms).

4.2.2 Instruction Set

Both organisms and IDOs possess complex instruction sets for tasks that allow them to maintain structure and continue to function through manipulation of their external and internal environments.

4.2.3 Ability to Act on Goals

This component of an intelligent entity is one which is hard to define. In organisms it is partially defined as life as an organism must be living to follow its instruction sets. In addition, this would include the fact that the goal set must not contain goals which an organism is incapable of reaching - a goal for a house cat must not be that it catch a 2000 pound Great White Shark. The ability to act on goals for an Intelligent Data Object requires that the IDO be part of the active IDO set of an IDOMS. It would also require that the physical implementation of the IDO be such that its goals are attainable. If an IDO is routed to nodes which don't exist in the network it will not be able to act on the goals which require such routing.

4.2.4 Ability to Obtain Environmental Information

Both IDOs and organisms have the ability to obtain information concerning the state of their environment. This ability is termed environmental sensing; discussion of environmental sensing abilities was given in chapters 2 and 3.

4.2.5 Goals

The goal set of an intelligent entity is perhaps the most difficult of the components to define. Many intermediate (proximate) goals exist in biological systems, with many combinations of these proximate goals leading to the same ultimate goals. Certain aspects of resources which must be obtained as proximate goals differ between IDOMSs and ecological systems. A discussion of the relationship of resources between these two systems follows.

4.2.5.1 Energy

Both organisms and IDOs require some source of energy to allow the instructions in the instruction set to be carried out. The instructions themselves are incapable of executing without this energy source.

Seeing the above relationships between biological organisms and IDOs, it would be helpful to examine in more detail the relationships and parallels between these two systems with an eye toward using principles which exist in ecological systems to suggest or define principles which might explain or predict the operations and behaviors of IDOs within computer systems.

4.2.5.2 Combining energy and matter into a single resource category

Energy and matter are often represented in Ecology as fundamentally separate components of the environment. Models often trace the passage of energy and matter through the ecosystem. Indeed, fundamental differences do exist between these two classes of resources. Energy cannot be recycled in an open system. An external source of energy must be supplied to a system to allow continued functioning of the system. Matter, on the other hand, can be recycled within a system closed to everything except the passage of energy. These differences are not important at the individual organism level on small time scales. Recycling of most matter requires long periods of time. In addition, the source of a essential resource is not important to an organism requiring that resource. In the long run, whether a resource recycles or not is of importance only in setting the limit as to the amount of the resource ultimately available to the organism.

For purposes of this study energy and matter will be clumped together as resources. Resources will be broken down into their distribution (and therefore availability) over time and space.

4.2.5.3 Differences between energy requirements of organisms and

IDOs.

Organisms require a constant source of energy to carry on biological functions. Some organisms (i.e. seeds, hibernating animals, etc.) have succeeded in getting their energy usage down to an extremely low level at times, but an energy supply is nevertheless always necessary. The energy that most organisms require can come from two different forms. A direct source of energy can be used (coming from the sun), or energy can be freed from chemical bonds (usually energy stored by some living organism derived ultimately from the sun). Thus although the source of energy to living systems is not constant over time, storage of the energy during peak times of availability (during the day, particularly during summer in temperate regions) allows for a steady if unequal supply of energy at all times of the day and year.

IDOs do not require a steady source of energy. They can sit without access to a CPU for hours or days without any loss of integrity. IDOs also do not have an apparent storage mechanism for CPU time. CPU time disappears with each tick of the clock, whether or not that time is productively used by a process. It is possible, however, that IDOs might have some mechanism whereby they can store access rights to the CPU (provided by some algorithm which increases a process's status whenever it is denied access, it might also be implemented as the earning of points which may be used later). Do to this volatility of CPU time, any comparison of an IDO system to energy pyramids of ecological systems is questionable at the present time (a system built on access rights may be built in the future - in such a case there may be a useful comparison).

A comparison of computer and ecological resource types grouped as to availability and stability is given in figure 4.2.

```
Clumped Availability <---------------------> Steady Supply
Volatile                                       CPU time
   |                                           oxygen
   |                                        carbon dioxide
   |
   |                      animal flesh
   |                      vegetation
   |  water                  water                      water
   |(deserts)       (forests, grasslands)   (rain forests,
   |                                     fresh water lakes)
   |   minerals                                   minerals
   |    seeds
   |nuts, fat stores
   |  solar energy ,data
Storable
```

Figure 4.2 - Stability and availability of select resources in ecological and computer systems

4.2.5.4 Aspects of resources.

There are four important aspects of resources relevant to this discussion: stability, distribution over time and space, abundance, level of refinement. These are discussed in the following paragraphs.

4.2.5.4.1 Stability

A stable food item is one which is nonperishable. An item with high stability will keep for a relatively long period of time. Examples of foods with high stability would be grains and nuts. Items with a low stability would not last long. Animal flesh is a good example of an item which has low stability in the warm temperatures.

Stability is important in that it allows for the evening out of an uneven supply of a resource which is needed on a more constant basis. A possible negative aspect of resource storage is the fact that another organism or IDO might utilize the stored resource to its own gain.

4.2.5.4.2 Distribution

The distribution of a resource is the measure of its variability over space and/or time. This aspect of a resource is important in that additional work may be necessary by an organism/IDO to obtain the needed quantities of the resource if it is clumped over time or space. If other resources are also clumped, the organism/IDO may require strategies which allow it to obtain all needed resources in an expedient fashion.

4.2.5.4.3 Abundance

If a resource (or access to a resource) is available in a limited supply, then competition is bound to arise among organisms/IDOs vying for that resource.

4.2.5.4.4 Level of refinement

The level of refinement of a particular resource is related to the amount of further investment which must be made to produce the resource required by the organism. In biological systems plants mine and concentrate minerals from the soil and air. Herbivores then eat the plants and obtain a more concentrated form of the minerals. The herbivores can thus do less work than the plants do to get sufficient quantities of the minerals they need (because they are more concentrated in the plants). In IDOs this might be realized as the obtaining of information from other processes which have already analyzed a large quantity of data to produce the information. The IDO is thus saved from a portion of the processing which would be necessary to derive the same information.

4.3 Comparisons

The purpose of the information existing in an organism is to cause that organism to behave in a manner which will increase its inclusive fitness, that is to increase the survivability of the organism's particular genotype. This increase can be affected in more than one manner. One way that the organism can increase its fitness is to produce offspring which will likewise produce offspring, etc. Another way in which fitness can be increased is to help produce related individuals' offspring. For this latter method to be productive, at least two offspring must be produced solely by the aid of the individual who forwent the production of its' own offspring if that individual is related to the offspring's parent by half in order that this strategy will be selective.

A main point which must be stressed here is that it is the information content of the instructions themselves which is being produced by the actions dictated by the instructions. The instructions' ultimate goal is therefore to produce more instructions. That is, instructions which are successful at duplicating themselves will persist through time where instructions which are successful at some other feat (and not at assuring the continued existence of duplicates of themselves) will not persist.

The instruction sets possessed by IDO's might also be considered to have the ultimate "goal" of reproducing themselves. If the instructions are successful at accomplishing the goal set forth for them by human programmers, they will persist. Otherwise they will be deleted and/or replaced by the human programmers in favor of a set of instructions which does in fact accomplish the proximate goal. In this way the instruction sets possessed by IDO's can be viewed as simplified but logical equivalents of the instruction sets encoded in the chromosomes of biological organisms.

4.2.5.2 Proximate goals of the instruction sets of organisms and IDOs.

At first glance the proximate goals of organisms and IDO's might seem quite different. Organisms in their day to day operations seek to survive and to mate and produce surviving organisms. At a lower level, they seek to obtain sufficient amounts of the matter required to build and/or maintain their physical structure, along with the energy required to process the matter in an efficient fashion and to move the organism itself if required, to gain the necessary matter and/or energy. IDO's, on the other hand, appear to exist solely for the purpose of gathering data for and transporting data to the operational entity responsible for its existence (some IDO's may be spawned by other processes, and will report their data to those processes, without coming into actual contact with the human(s) ultimately responsible for (and profiting from) their existence). With a more indepth examination, we see that IDO's must have a data set which takes in to account the fact that the IDO must survive to complete the mission of gathering and reporting data, that the IDO must complete this mission in a timely fashion if the mission is to be considered a success, and that the actions of the IDO must not jeopardize the success of subsequent IDOs as they venture forth to accomplish related objectives. (An IDO which must receive information from a clerk and succeeds in obtaining the required information in a timely fashion through questionable means such as harassment, etc., might cause the next IDO with the same origination to incur a longer than average waiting period, and/or lesser quality service). In addition, absolute survival of the IDO may not be an optimal strategy for obtaining information in a timely fashion. IDO's will likely work in classes where IDO performance will be measured as a class statistic rather than as individual performance. Pulling stops which might serve to prevent an untimely fate of an individual IDO in favor of improved system performance might well be necessary to increase total information flow. Other IDO's with different instruction sets might be dispatched upon the demise of a data-gathering IDO of a particular class in order to retrieve as much of the information already gathered by that IDO as possible. This approach might well reduce the ill will that can be created through multiple requests for the same data (where such data consists of authorizations provided by humans with limited time), or where the data previously collected consists of one-of-a-kind data which can not be easily replaced. To facilitate the regathering of data from disabled IDOs, external "paper trails" might be kept at the nodes where the IDOs visited, with information stored, possibly in coded fashion, on the data gathered at the present node along with information as to where the IDO had been and was going, to allow for retracing the trail and gathering the data already collected. To prevent these paper trails from crowding the system, they should probably be dated and removed after some relatively short length of time (where the definition of short would depend on the class of IDOs involved).

In this chapter an abstract intelligent entity and its environment have been defined in terms of component parts. The resultant model represents both biological organisms and IDOs operating in ecological systems and IDOMSs, respectively. A demonstration of the applicability of this model to both of the modelled systems has been given. The abstract intelligent entity serves as a common ground upon which either of the two modelled systems can serve as a model for the other system. In the next chapter, organisms operating in ecological systems will be used to provide principles of interaction which can be applied to the development of efficient IDOMSs.

Chapter Five

Selected Ecological Principles

The set of ecological principles which may have immediate application to nondeterministic concurrent processing software systems has been formed from a subset of the known ecological principles. Principles dealing with predator/prey relationships (where both predator and prey would be represented by IDOs), energy pyramids, and others seem to have no immediate correlates. Those principles which might have application in the concurrent processing area are too numerous to be discussed in a thesis of this scope. Several principles have been chosen which in the author's opinion have potential in describing concurrent system principles.

A general discussion of the types of interactions possible among organisms is presented. A discussion of the selected principles is then provided in terms of their utility in describing and predicting the behavior of IDOs.

5.1 Types of Interaction between Two Species

Populations of organisms can have positive (+), negative (-), and neutral (0), effects of interactions with other species. These effects should be classified as changes in numbers of individuals in the population or reproductive success of the species, and not necessarily as good, bad, and no effect. Some small animal species, for instance, are controlled by predation. If the predatory species were removed from the population, the smaller animal (voles, etc.) population might overpopulate, thereby causing a much greater bad effect on the the population. Interactions between populations of two species may be of six basic types which correspond to combinations of 0, -, and +, as follows: 00, 0+, 0-, ++, --, and +- [Odum83]. Some of the combinations can be broken down further to produce nine important types of interactions. These interactions can be summarized as follows [Odum83]:

1. neutralism (00) - neither population is affected by association with the other

With neutralism, no effective interaction exists. In organisms, this may be represented by plants living in different subhabitats, and in animals, by two species which have no niche overlap.

In the case of intelligent data objects, this situation may be represented by IDOs which run in different time frames, or by IDOs which have different data requirements in a system where CPU time is not limiting.

2. competition, mutual inhibition type (--) - both populations actively inhibit one another

Species which inhibit each other's reproductive success without direct competition for resources would fall into this category. This type of competition is found more among higher level animals as they can exhibit behavioral displays. [Odum83]. Examples of behavioral displays being used in this manner include territorial animals which defend their territories against other species, thus limiting resource use for more than one species (assuming all species involved are also territorial).

No IDO interactions of this type are immediately apparent.

3. competition, resource use type (--) - both populations adversely affect each other indirectly through use of the same limited resources

This type of interaction is seen in organisms where two species share a common, limited resource. Examples include plants competing for light, animals competing for limited plant or animal food sources, etc.

IDOs operating in an overloaded system compete for CPU and/or main memory, IDOs requiring the same data from a data base compete for that data, etc. Worm programs which search out idle computers from within a computer network, and occupy all idle computers found (up to some maximum number) [Shoc82] compete at the network level with any other such worm programs. At the individual machine level, these programs might be the ultimate competitor for the machine with the machine operator when they possess 'high strung' behaviors, as one worm described by Shoch did when the instruction set was inadvertently changed as it ported itself around a network.

4. amensalism (0-) - one population is inhibited, the other not

affected

An example of biological amensalism which is often cited is the production of poisonous by-products by one plant which limit the growth of another species [Bego81].

A busy-wait by a high priority IDO in a priority queue for CPU time could be thought of as being in this category. The high priority IDO would not derive any gain from tying up CPU cycles, while any low priority IDO would be indefinitely delayed.

5. parasitism (+,-) - one population adversely affects the other by direct attack but is dependent on the second population

Many parasites exist in biological populations, including tapeworms, protists, fleas, etc.

In computer systems, we would like to think that parasites don't exist. The potential remains, however, for IDOs which pirate information from other IDOs for the benefit of their creator. In such a case, the "bread line" would quickly dry up if the effect was so great as to remove the IDOs which are being pirated from the system (bankruptcy of companies, etc.).

6. predation (+-) - one population adversely affects the other by direct attack but is dependent on the second population

Predators in biological systems are well known to the general population. The difference between predators and parasites is a matter of degree. Parasites usually do not kill their victim while predators do.

Predatory IDOs in computer systems might destroy IDOs to benefit their creator. This benefit might derive from the information pulled from the destroyed IDO. More likely, however, is the potential for adversely affecting a company such that money can be made through stock purchases or sales, increasing the value of options, etc.

7. commensalism (0+) - one population is benefited, the other not affected

Many examples of this type of interaction exist in ecological systems: birds feed on insects kicked up by feeding cattle, aquatic organisms live in ponds made behind beaver dams, etc.

One can envision IDOs which would interact with this type of behavior pattern. IDOs constructed such that important system information could be extracted from these IDOs as they wait in a queue for a database resource would help other IDOs dependent on the extracted information to operate efficiently in the system with out losing any time or resources themselves.

8. protocooperation (++) - both populations benefit by the association but relationships are not obligatory

These types of interactions are believed by some to be precursors in evolutionary time of mutualistic interactions. An example is certain crabs which carry coelenterates on their backs, thereby deriving protection from the stinging cells of the coelenterates. The coelenterates are transported about and derive benefit in the form of bits of food left over when the crab feeds. Both species are able to live without the other, though each does better when they have this association.

IDOs interacting is this fashion could be designed such that they could communicate to each other the ownership of common commonly-used goals, with some method being used to share the work and benefits of reaching the goals. One possibility might include being transported together to save on overhead, thus helping both IDOs hold down processing costs.

9. mutualism (++) - growth and survival of both populations is benefited, neither can survive under natural conditions without the other

A well known example of mutualism consists of certain species of ants and acacia trees which coexist in Central America, where the ants protect the acacia from harm, while the acacia provides a safe home for the ants.

IDOs with similar goals might be bundled together into one super IDO which divides goals at nodes, thereby increasing the efficiency of processing. Back at the home node, the IDOs would be separated, common data needs would be shared, and each IDO would be returned to the originator. Such bundling could be made obligatory such that a single IDO would not carry the required information necessary to fulfill all goals.

"The interactions listed above are not all equally important in all communities. Odum [Odum83] lists three principles based on the above categories of interactions which he felt worthy of emphasis. They are:

1. Negative interactions tend to predominate in pioneer communities or in disturbed conditions where r-selection counteracts high mortality.

2. In the evolution and development of ecosystems, negative interactions tend to be minimized in favor of positive symbiosis that enhances the survival of the interacting species.

3. Recent or new associations are more likely to develop severe negative co-actions that are older associations."

These principles can also be applied to associations of IDOs which might be found operating within operating systems and networks. The application of these principles to computer systems will be discussed below.

The first principle states that negative interactions predominate in pioneer communities or in disturbed conditions. Early operating systems were often found to be in error when jobs run on the system did not produce expected and/or reducible results. Database anomalies were discovered from errors which occurred in working systems [Codd72]. In general, systems which have recently been constructed from pieces brought together in a new association have a greater likelihood of negative interactions among the pieces. We speculate that any system which can not guarantee that a IDO will not suffer from indefinite postponement or destruction can expect to have greater competition for resources (and thus negative interactions) due to the necessity of introducing multiple copies of IDOs to insure that at least one of the IDOs will safely return to home base within a reasonable amount of time.

The second principle is realized in computer systems as bugs are discovered and fixed, thereby increasing the reliability of the system, which in turn decreases the need to duplicate effort, which lowers system load and increases overall productivity while decreasing negative effects of one IDO on another.

The third principle is important to the development of computer systems. It states that new associations are more likely to have severe negative interactions than older associations. This principle is of importance in that it states what is already known about new computer systems, i.e., that they can be expected to have bugs in them which will cause problems. Since there is no history of system behavior to use in predicting future performance, no IDO can afford to behave in any manner which might prevent it from reaching its goals (usually obtaining a finite set of resources within some amount of time), especially where such a behavior might help the overall system performance with little or no return to help satisfy the goals of the IDO itself. This problem might be simply stated as "every IDO for itself". Later, as a history of performance is developed, and reasonable protocols for IDOs operating within the system are developed through software development, behaviors can be programmed in to IDOs to cause them to operate in less negative fashions.

The same types of principles dealing with associations of organisms may also apply to associations of IDOs in computer systems. The methods by which groups of organisms increase cooperation might therefore prove useful for increasing system performance of new associations of computer users and/or IDOs.

5.2 Optimal Foraging

Optimal foraging theory seeks to define a set of decision rules which an animal should follow if it is to optimize its food intake. Krebs [Kreb78] discussed these rules in terms of general strategic rules which apply to a wide range of animals. The rationale given for examining the possibility of the existence of optimal foraging in animals is this: "animals will, as a result of evolutionary selection pressures, tend to harvest their food efficiently. Thus one can work out in theory the decision rules which would maximize the animal's efficiency, and these rules ought to allow us to predict how the predator makes its choices [Note that the words 'decision' and 'choice' are not intended to imply anything about conscious thought, they are simply a shorthand way of saying that an animal is designed to follow certain rules. The term 'predator', is used to include more than just carnivorous animals: to a seed, a finch is as much of a predator as is a shark to us.]" Krebs goes on to point out that efficient food gathering could be measured in a number of different ways, including maximizing food intake, maximizing vitamin intake, minimizing the chance of being killed by a larger predator, etc. The models to be discussed here will deal only with how an animal can increase its net food intake once it has decided to forage.

5.2.1.1 Prey Choice

Krebs divides the choices to be made by an actively foraging animal up into three main groups: which types of food to eat, where to hunt for food, and what type of searching path to use in hunting for prey or foraging places. Of these three types of choices, the latter two are unlikely to produce currently usable theory for intelligent data objects, i.e., the physical location of the "prey" item is usually known. The latter two types of choices deal solely with the location of prey items (not with choosing an optimal path to get to a located prey item). I will therefore limit the discussion here to which types of food an animal should eat.

A prey item eaten by a predator has a cost in terms of the time associated with subduing and eating the item. A benefit is also derived and can be measured in terms of net energy value (the energy obtained above and beyond the energy costs of obtaining it). If a predator is to optimize its gathering of prey items, it must be able to detect the profitability (net energy value/handling time) of any given prey item. Studies have shown that predators are able to choose profitable prey items, and that they do so regularly (Figure 5.1).

Choice of prey items becomes less black and white when search time is added into the cost/benefit analysis. If a predator chooses only the most profitable food items, it may starve while searching for another profitable item if the most profitable items are rare. The actual distribution of prey items in the diet should be dependent on the density of the best prey items along with the density of any other prey items which might be consumed. Obviously, if the best prey items are sufficiently dense as to allow their capture in the same amount of time it would take to capture any other prey item, only the best prey items should be consumed. If on the other hand, the best prey items are few and far apart, then some percentage of lesser prey items should be included in the diet. All prey items can be ranked as to their cost/benefit ratio (as figured for energy expended on handling and digestion as related to energy and/or matter benefit derived). The prey items which are to be taken by a predator are determined by calculating which items on the list will cause a net decrease in overall benefits derived if that prey item is taken. All prey items lower in the list will then not be advantageous to take. Graphs which Krebs modified from the work of others are presented to show the relationships represented in optimal prey choice (Figure 5.2a & b). This optimal diet model was first presented by MacArthur and Pianka (MAC 66) in 1966 and has been presented in several different forms since that time. The important predictions which arise from this model, as presented by Krebs, are as follows: "predators should (i) prefer more profitable prey, (ii) be more selective when profitable prey are common, (iii) ignore unprofitable prey which are outside the optimal set regardless of how common they are".

The model of prey selection given in Figure 5.2, a & b, does not represent the additional time which would be necessary to reject or accept any prey item. In situations where many unprofitable prey items are present, the processing time required to reject all unprofitable prey will become a major aspect in prey selection decisions. In this case, the E/T line (net food intake per unit time) should rotate clockwise, causing the predator to take more of the less profitable food items than it would if recognition time were not a factor (Figure 5.2c). With this addition, to the model, any significant recognition time required by an animal in making choices should cause the removal of the third prediction of the earlier model. Tests of this model using four different species of organisms show that it is indeed applicable [Kreb78].

5.2.1.2 Optimal foraging for patchily distributed foods.

When food items are found in patches, additional foraging techniques may be applied to maximizing food intake over time. The decision rules discussed above should apply to processing of prey within any given patch. Additional decision rules are needed to determine which patch to forage in (if patches vary in quality). Intuitively, one realizes that a predator should be able to optimize its food intake if it spends more time foraging in high quality patches over low quality patches unless high quality patches are rare. Part of the difficulty in choosing patches is determining whether or not a patch is of high or low quality. Usually making such a determination requires spending some time in a patch. The question of how long a

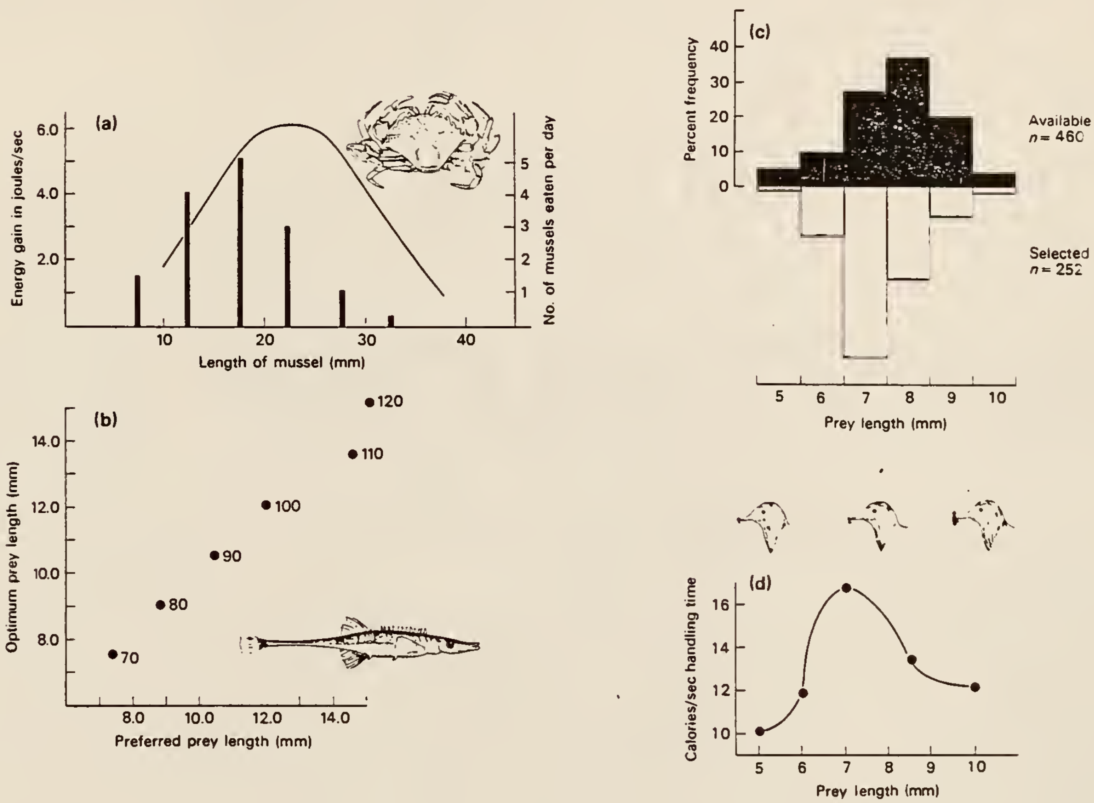

Figure 5.1 - "Predators choose profitable prey. (a) Histogram of numbers of mussels of different size classes eaten by 6.0-6.5 cm. shore crabs with unlimited prey. The curve shows the energy gain per unit prey breaking time" [Kreb78;Elne78]. (b) "Preferred length of Neomysis integer eaten by different sizes of 15-spined sticklebacks in the wild, plotted against optimum prey length (determined by dry weight of prey/handling time). The figures by the dots refer to fish size (length in mm)" [Kisl76;Kreb78]. (c) "Selection of flies by pied wagtails. The upper histograms show the available and preferred distributions of flies, the lower curve, (d), shows the profitability of different sizes of prey in calories per second of handling time" [Davi77;Kreb78].

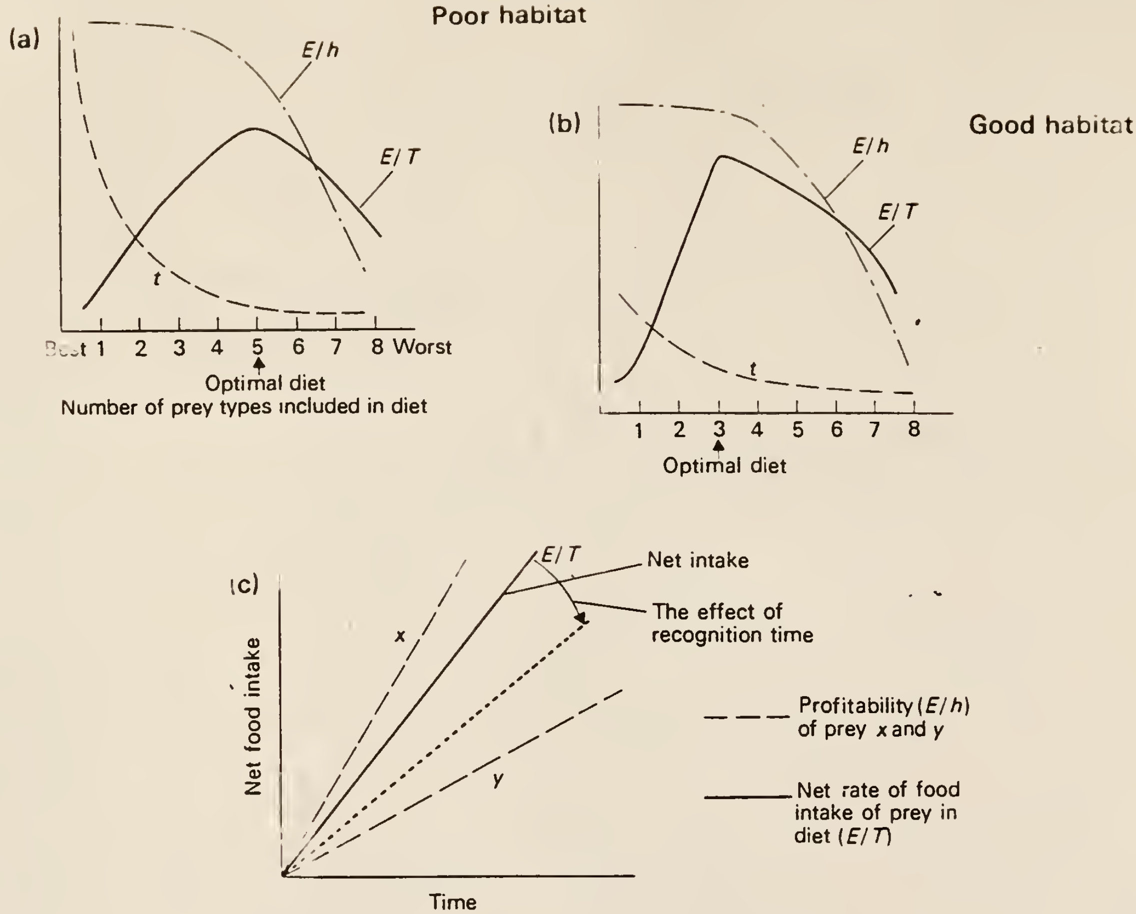

Figure 5.2 - A graphical model of optimal breadth of diet. (a) & (b) show, for a poor and a good habitat respectively, the consequence of including different numbers of prey types (ranked in order of profitability) in the diet. As more prey are added, travel time (t) decreases, but so does the average profitability of prey eaten (E/h). The curve of net food intake per unit time (E/T=E/t+h) rises to a peak and then declines. The peak is the optimal diet breadth: in the poor habitat, where good prey are scarce so that travel time is long, the optimal diet includes more prey than in the good habitat. (c) If the predators net intake is E/T (solid line), it should not eat any prey such as Y with a value of E/h below the solid line. Even if prey such as Y are very common, the slope of the solid line cannot be increased by adding these prey to the diet. On the other hand a prey type such as X should be eaten as it increases the value of E/T. If prey such as Y have a recognition time, they might, when common, lower E/T (solid line slope reduced to dotted line) [Kreb78].

predator should remain in any patch can be divided into three questions which are dependent on the patch type. These patch types can be defined as i) patch types which don't change in quality over time, ii) patches which are depleted as a result of the predators activity, and iii) patches which change in quality independently of the predator's activity. Krebs discusses the implications of each of these patch types on a predator's strategy. This discussion will be limited to the third type of patch, as it is this type which will most likely be immediately applicable to IDO foraging strategy.

Patches which change in quality irrespective of the predator's presence are not as common as those which are depleted due to the predator's presence. Changes not due to the predators presence can occur due to the seasonality of a resource or due to diurnal changes in resource concentrations. Another cause can be due to movement of a patchy resource. In a study of spotted flycatchers (Davi77), swarms of insects represented relatively unlimited resource patches, and their movements relative to perches affected a geographical patch around a flycatchers perch. Field studies found that flycatchers switched perches whenever an intercatch interval was longer than 1.5 times the average intercatch interval. In such cases the swarm had usually moved away from the perch. The decision to move in this case is a tradeoff between the chance of leaving while a swarm was still within striking range, and the cost of waiting a long time before the next swarm arrives.

5.2.1.3 Optimal return times to renewable patches

Many animals utilize resource patches which are renewed at relatively rapid rates. In utilizing this type of resource, an animal must time its return rate so as not to return before the the resource is completely replenished. At the same time, the predator does not want to wait too long to return, or it may spend too much time traveling. Returning too early or late will thus decrease the animal's foraging efficiency [Putm84].

Charnov et. al. [Char76] point out the fact that reutilizing renewable resources requires a condition such that the predator need not worry about other predators getting to the resource before it returns. This type of resource utilization is thus usually found in territorial animals or in species that forage in groups.

Territorial nectar-feeding species are good examples of species which utilize this foraging strategy [Putm84]. Field work done on sun birds and honeycreepers indicate that they forage in a manner which is geared toward the nectar renewal rate in the flowers they visit [Gill75,Kami79]. Davies [Davi76] demonstrated the use of this type of foraging strategies in Pied Wagtails which feed on insect debris washed up on river banks. They patrol the river bank by systematically moving up one river bank and then down the other. Davies demonstrated that the pattern of the search served to maximize the birds return to previously depleted areas. He also showed that the birds expended greater effort on evicting other birds from their foraging area than other members of the same species did when feeding in pastures [Davi81].

5.2.2 Application of optimal foraging ideas to foraging by IDOs.

IDOs forage in the sense that they must gather data and/or authorizations from databases and/or human operators at workstations. Foraging in IDOs differs at least superficially from foraging by animals in several aspects. First, IDOs usually have specific pieces of data or authorizations which they must obtain, they cannot (at least at this stage) make a choice therefore between different "individuals" of some type of data. Also, once a single piece of data is obtained, no more data of that type is required. Choices among several different types of data are also not usually available, as the creator of a IDO usually specifies all the data which is to be obtained, usually with no facility for making substitutions. The cost of obtaining a given resource can be measured in terms of CPU time, real time, or some charge made by the supplier of the data or authorization. Since a monetary charge will probably not vary with the number of times an IDO considers queuing up to obtain a resource, the first two cost estimates are probably more applicable to any discussion of optimal foraging in IDOs. Of the two, real time is probably of greatest importance as companies try to meet deadlines, get the edge on the competition, etc.

Current foraging techniques by IDOs consist of following a list of data to be obtained, with the possibility of following several different paths based on the idea that some data may be obtained in parallel. Choices between these paths would currently be made in terms of optimal path choice (i.e., the traveling salesman problem). Some discussion has been generated on the possibility of setting a maximum waiting period for any given resource. If this period of time were exceeded, the IDO would dequeue itself, and obtain other resources at the the same level before retrying to obtain the current resource [McBr87]. The advantages of programming an IDO to display this behavior would be that it would allow an IDO to detect an unreasonably long waiting time (an operator might be on lunch break, etc.) and optimize its collection of resources by changing the order of retrieval. The disadvantage of this system would be twofold: 1) the IDO may dequeue itself just before obtaining the resource, thereby necessitating another wait in the queue to obtain a resource which was within grasp, and 2) time must be spent in the queue to determine whether or not the queue is unreasonably long. Optimal foraging theory might well be used to suggest alternative methods for optimizing the retrieval of data by IDOs.

5.2.2.1 Changing the model

To apply optimal foraging theory to IDOs, some changes must be made in the way we model the environment within which IDOs operate. As noted above, a purely physical model in terms of location of data servers, operators, work stations, etc. will likely prove unsatisfactory in meeting our goals. A usable model can be obtained, however, by extending our model to model the location of resources in both space and time. Such a model can be represented by a highly connected space/time graph. In addition, cost functions for any given resource must be specified to allow us a method of evaluating an optimal foraging model.

By adding a space dimension, a single resource which must be obtained becomes a number of possible resources from which a choice must be made. An IDO at any given node may decide, for instance, to enter a queue to obtain a resource at the present time, or it may travel to other nodes, possibly obtaining other resources, before returning to the current node, at which time it can again make a decision to enter the queue or go elsewhere. The physical resource at the current node thus becomes an infinite number of resources with associated cost functions.

The cost of the resource in the current node might be represented as the real time which must be spent waiting in queue, if that time is much greater than the average time a IDO has to wait in the queue, it might be more profitable to go elsewhere, obtain another resource, and then return to the current queue, with the hopes that the queue length will have decreased in the ensuing time. In such a case, the cost of the travel to the other resource and back must be added in when determining the future cost of the resource.

The available instances of a needed resource extend out in all directions infinitely, with the cost of the resource tending to increase as the IDO moves away from the current node due to the increased travel cost. This increasing cost function will in some cases be locally decreasing, due to the fact that the initial cost of the resource being high if the queue is much longer than average. In such cases, one would expect that the resources surrounding the current resource in the time/space graph would tend to be declining in cost, until the length of the queue gets back closer to average length, in which case moving still further from the initial resource will result in gains in resource cost (due to travel time).

Patchiness of the resource results when an IDO obtains several other resources before returning to a resource which had earlier been high priced. The costs associated with the new queue size and its associated options might be thought of as a separate patch from the costs which would have been present at the earlier time.

Relation of costs in IDOMSs to those in ecological systems is necessary if optimal foraging is to be utilized in IDO design. The benefit of a given resource can be thought of as fixed. The costs of a resource are therefore solely responsible for varying the cost/benefit ratio for a given resource. These costs can be divided as follows: handling time/eating time for animals can be represented as the time which is spent waiting in a queue for an IDO. Searching time in animals correlates with the traveling time involved in getting from one temporal location of a resource to another temporal location of the same resource. Recognization time for an animal (the time spent identifying a potential prey item) would be represented by the time required to request to be queued (probably negligible in most instances).

Animals are able to get information from their environments to allow them to make the decisions necessary to forage optimally. It is necessary to assume some capacity for IDOs to do the same in computer systems. Specifically, IDOs should be able to request and obtain information on queue size, and should be able to determine in some manner the cost of travel between any two temporal instances of a resource (at least an average or weighted average cost). It would also be helpful for an IDO to know the system load and the variance of the load, as this may affect the profitability of seeking another resource instance (in a system with a higher load then normal, other "nearby" instances may not be cheaper than the current instance).

The choice that an IDO must make in foraging is whether to take the current instance of a resource or whether to wait and take it at some future time. The advantage of taking a resource immediately when at the node where the resource is located is that no travel time must be added in to the effective cost of the resource. The disadvantage of taking a resource immediately is that there may be a high cost in terms of time spent in obtaining the resource, time which might be better spent collecting other resources at other nodes (or possibly at the same node). The average cost of obtaining any given resource should be calculable. If this cost is kept by the system such that it may be accessed by an IDO, the IDO may be able to make a choice as to whether it should leave or stay. If the resource desired by the IDO is a queued resource with constant availability, it should be easy for an IDO to calculate the relative profitability of staying or leaving. If, on the other hand, the resource is a queued resource with spotty availability, say a human operator or workstation where there may or may not be someone servicing IDOs at any given time, the calculation may be more difficult or impossible.

In the case of a data server which is constantly available, an IDO will do better to leave if the current queue length (measured in real time necessary for processing) is greater than the average queue length by more than the time it would take to travel to another resource and back (assuming this does not disturb an optimal path). If, on the other hand, the cost of travel is greater than the difference between the current queue length and the average, it would be preferable for the IDO to take the resource at the current time, even though the current resource is not at the optimal cost. In short, the IDO should take the resource with the least cost in the space/time graph.

Decision-making in choosing among resource instances which are nearby in the space/time graph are fairly straight forward. Any possible path away from and back to the current instance must be examined for the possibility of a cheaper instance of the resource. If the cost of travel is non-negligible, this process should not involve very long path distances before the costs begin to rise above that of the current instance. A problem arises, however, if one considers resources where the time needed to process a single IDO by a data server is non-negligible. In this case, any resource instances in the same area of the space/time graph will be functionally related in terms of queue length and therefore cost. In this case, it may be better for the IDO to get a significant portion of the other resources required before returning to retry for the current resource (in the hopes that the queue length will have fallen closer to average length). In this case, resources of one type which are close to one another in the space/time graph may be considered to form a patch, with the cost of any resource within the patch being a good indicator of the overall quality of the entire patch. By looking at these resource groupings as patches, one can utilize optimal foraging theory on patch use to suggest techniques for minimizing the cost of such resources.

5.2.2.2 Optimal return times to resource patches for IDOs.

Optimal foraging literature may also prove useful in calculating optimal return times to renewable resource patches in computer systems. Although IDOs are not responsible for the depletion of a given patch of resource items, the return time theories should prove helpful in building a workable theory of optimal return times for IDOs.

As stated earlier, the possible values of nearby resource instances in the space/time graph is closely related to the value of the current resource instance if the processing time for a single IDO is non-negligible. In such a case, an overloaded queue will take some time to return to average queue length. In this case, it is probably preferable for the IDO to collect a number of other needed data items before returning to the current physical resource instance. Just what that time will be will depend on several factors: the processing time for a single IDO, the queue length, the distance that must be traveled to process N number of other resource items, and the physical patchiness of the other resource items. Additional factors which would be more difficult to account for but might be equally useful in determining return time could include system load, average load, and current variance of system load, and historical data on resource use of that resource, especially in regards to patterns in terms of time.

5.2.3 Summary

Optimal foraging techniques have been well described in ecological literature. Although many of the possible aspects of optimal foraging do not appear to be applicable to foraging of IDOs in IDOMSs at this time, at least several areas of theory do exist which might be usefully applied towards describing and defining optimal foraging techniques for IDOs.

5.3 Sociobiology

Sociobiology is "the application of evolutionary biology to the social behavior of animals" [Bara77]. The goals of this relatively new area of science are thus the explaining social behaviors in an evolutionary context. If organisms possess behaviors for some long period of time (as a species), then those behaviors must have a positive or neutral effect on those organisms which exhibit them. Social behaviors which appear to benefit organisms other than the owner of those behaviors therefore must also have a net benefit or at least no net cost for the organism displaying them.

Some areas examined within sociobiology include parent/offspring conflict, altruism, mate selection, parenting, social competition, and spatial competition. Since IDOs operate in an environment including other IDOs, at least some of the theories from sociobiology should be useful to increase our understanding of IDO interactions.

For full scale application of these ideas, it is imperative that the same model of instruction inheritance and relatedness apply to both organisms and IDOs. Although our model does not cover this aspect of the organism-IDO relationship, one of the conditions for use of IDOs is that the integrity of the IDOs is assured [Unge87]. With this assurance, it is reasonable to assume that the concept of relatedness can be applied to IDO instruction sets.

The instruction set of an IDO can be thought of as having the sole purpose of furthering the goals of that IDO. At first glance, these goals would seem to consist of collecting a given amount of data/authorizations in the least amount of time possible. Sociobiological theory would say, however, that the goal set is actually hierarchical in structure, and that each individual instruction will affect all levels of this goal hierarchy simultaneously. An instruction set which is written without consideration of any but the most immediate goals of the IDO will nevertheless have an effect on goals at other hierarchical levels.

A better understanding of a goal hierarchy can be obtained through examination of the goal hierarchy of an organism. At the lowest level, an organism "wants" to maximize its energy and matter uptake to use ratio (within the bounds of its needs) while minimizing its chance of being injured or dying. At a slightly higher level, and organisms "wants" to maximize its reproductive fitness by maximizing its representation in the next generation (a rough estimate of this would be the number of offspring the organism has). At a still higher level its goals would consist of maximizing its inclusive fitness (kin selection) [Wils75,Bara77,Curt83]. Higher levels of goals which include still larger groups of organisms have been proposed [Wynn62]. These levels would include group selection [Wils75,Bara77], instructions which work toward the good of the species, and instructions which work to help assure the survival of other species. The mechanism proposed for these higher levels of selection has usually involved differential selection between groups of organisms on up to entire ecosystems. The general consensus at the present time holds that selection at these higher levels doesn't exist [Wils75,Bara77]. For selection to operate at these higher levels, it would be necessary that populations of organisms have extinction rates of populations which are comparable to death rates of individual organisms. Such levels of population extinctions do not exist, therefore individual selection effectively stops any group selection which might otherwise occur [Wils75]. Figure 5.3 shows a generalized goal hierarchy for organisms and IDOs.

organism	IDO
effective foraging for resources	effective foraging for a datum or authorization
production of as many surviving offspring as possible	obtaining all required data and authorizations
production of as many genes in the next generation as possible (kin selection)	set of IDOs working toward a common goal (may include forking and/or a set of IDOs accomplishing some goal
species survival (nonexistent in nature)	success of all IDO's used by one department
ecosystem survival (nonexistent in nature)	success of all IDO's used by one company

Figure 5.3 - Relationship of goal levels between organisms and IDOs.

Some of the goals which might be desired in IDOs (efficient use of the company's computer system, etc.) are not effectively selected for in biological organisms. Biological organisms have no controller which determines whether or not the organism will continue to exist. Only natural selection, which operates at the level of the individual, is available to determine whether and instruction is effective or not effective. Thus, for higher level goals than the immediate goals of an IDO, some control must be exerted from above the single IDO level to insure that the higher level goals are met. This type of control (IDO integrity) is a fundamental part of the definition of an IDO [Unge87]. If for some reason this control is not exercised, IDO selection may be at a level lower than that desired. In such a case, it may be useful to extend the model to cover goal inheritance and relatedness (not covered by the current model).

The idea of a goal hierarchy is not a new idea in the computing and information sciences field. Similar ideas have been recognized, studied, and utilized for years in areas of computing sciences which include software engineering, operating systems, network design, and database construction. By utilizing a fresh perspective of looking at how another discipline views and categorizes goal hierarchies, valuable insights may be gained on the effective construction of goal sets. In addition, goals which are unattainable in current systems may be discovered.

5.3.1 Altruism and Kin Selection.

An apparent paradox which has been recognized for years in the field of ethology is the presence of apparently altruistic behavior in individuals of many species. Mothers will risk their lives for their children, siblings will defend siblings, and in some cases, apparently unrelated individuals will give alarm calls in an attempt to warn other individuals (sometimes of other species) of the presence of a predator. Such behavior was long thought to increase its owner's risk due to the increased risk of detection by the predator.

With the application of evolutionary theory and scientific methodology to behavioral studies, it was shown that apparently altruistic behaviors actually increase the inclusive reproductive fitness of the individuals displaying them [Bara77,Wils75]. Parents and siblings are related to each other by one half, thus, if an individual can save two or more of its siblings/offspring through its own death, it will be selected to do so [Bara77,Wils75,Curt83]. This concept of relatedness can be extended to all blood relatives which can be recognized by an organism (usually only members of the immediate family group).

It has also been shown that individuals giving alarm calls are not true altruists. Alarm calls are usually given in a high pitch which is difficult for predators to localize. Predators often rely on stealth and concealment to aid them in their hunt. Thus when the predator has been detected, it is often better for the predator to move on to hunt elsewhere, thereby decreasing the risk for the individual(s) which sounded the alarm [Wils75,Bara77].

5.3.2 Relatedness in IDOs.

The concept of relatedness is a useful one in IDO design. IDOs which are related to an IDO are from the same project, department, division, company, or corporation. An IDO will achieve more benefit from helping another IDO per unit of effort if that IDO is more "related". Therefore, an IDO will receive more benefit from helping an IDO from the same project then it would from helping an IDO from another division. These concepts should be considered when deciding on the level of "niceness" to be programmed into an IDO. IDOs from the same project are probably more apt to be running on the same computer than other, less related IDOs. Thus "nice" behaviors will increase the productivity of all IDOs in the same computer. Similarly, if an IDO can recognize an IDO from the same project at some distant site, it should give priority to that IDO if it (the other) is running short on time necessary to meet its goals.

Many theories of sociobiology have been quantified, and may be of use in planning the interactions of IDOs in a large network.

5.4 Security

Security in computer systems might be broken into two areas, security of the system as a whole from outside intrusion by programs bent on wholesale destruction or gradual degradation of the computer system. One reason that a programmer would try such unethical practices might be the desire to bring a system down as a challenge. At the other end of the scale, a programmer might attempt such a practice for personal financial gain.

Security of the system as a whole is not a problem to which the ecological literature can easily be applied. Many cases have been documented in which alien species are introduced into an ecosystem. Often, these alien species are able to cause great disruption of the system before counteractive forces are able to begin to establish some control mechanism. Since the properties of the ecosystem arise out of the behaviors and instructions of the organisms themselves, introduction of an organism which possesses an instruction set developed in some other ecosystem can cause great effect on the system to which it is introduced.

The biological systems which might provide the most insight to protection from outside intrusion are organismic systems. These systems consist of the integration of a number of systems within a single organism. This type of system might best be compared to the operating system of a single computer, operating to maximize as much as possible the goals of the users of that computer. The principles noted might also be extended to a small network operating system which is well integrated and constructed by a single software development group.

5.4.1 Security in Biological Systems

A summary of the system of an organism which provides security from invasion by other organism can be found in any good basic Biology text. The following account has been derived from Curtis [Curt83].

The first order of defense in an animal is the skin and mucous membranes. They represent barriers to foreign invaders such as viruses, bacteria, parasitic insects, arthropods, annelids, etc. The skin and mucous membranes are sufficiently impenetrable as to keep out the vast majority of invaders.

The second order of defense in animals is the inflammation response. Should the outside barrier of the body be broken, histamine and other chemicals are released, causing an increase in the flow of the blood to the area. Specialized white blood cells flock into the tissue of the injured area and engulf foreign microorganisms, viruses, and other matter.

Should the first two orders of defense be insufficient in keeping foreign invaders out, two other defenses are used. The first, the release of interferon by cells infected with a virus, tells nearby cells to increase their defenses against the invader. The second, the immune response, is much more general in its attack, and is described further below.

The immune system includes white blood cells of two types, T lymphocytes (T-cells) and B lymphocytes (B cells). At any given moment, approximately 2 trillion B lymphocytes are circulating in the human body. These lymphocytes represent the ability to recognize a vast number of antigens (usually foreign protein molecules). Each individual lymphocyte can recognize one type of antigen, and only a few of that type of lymphocyte exists in the body. Should any of these lymphocytes come in contact with the antigen for which it is specific, it begins active metabolism, and begins to divide, producing two types of daughter cells, plasma cells and memory cells. These cells differ primarily in longevity, with the memory cells surviving periods up to the life of the individual. If the foreign antigen was part of an invading virus or microorganism, it is likely that many more are present in the body. The production of daughter cells which divide further themselves and produce antibodies is therefore a defense against the invaders. The antibodies produced can have three types of actions against the invaders. The can coat the surface of the invader, promoting its ingestion by a prowling white blood cell. They can combine with the antigen and interfere with its action. Finally, they can lyse (cut open) a foreign cell in conjunction with the complement in the blood. Once the invasion is squelched, the plasma cells slowly die off and the memory cells remain in relatively high numbers, ready to respond quickly to a second invasion, thereby giving the human "immunity". The T-cells respond similarly, but act against the bodies own cells by identifying cells which are harboring viruses, cut them open, and expose the viruses inside to the action of the antibodies.

Disorders related to this defense systems include allergies and autoimmune disease. Allergies are potentially harmful responses to nonharmful antigens, while autoimmune diseases are the result of the immune system "forgeting" the identity of some of the bodies own cells, thereby prompting an immune response against the body.

The price of invasion is so high, and the benefits so great for the invader, that a costly system to ward of the effects is essential to guarantee continued operation of the system (the organism in this case).

5.4.2 Applications of Immunological-like Security Measures to IDOMSs.

Applicability of animal defense against internal attack is unclear. The first line of defense in which microorganisms, viruses, etc. are not able to easily penetrate the system is obviously already in place in most computer systems. Such defenses include users access rights to the system and to memory within the system. Secondary defenses might also be considered to be in place with monitoring of users' use of the system, with loss of privileges awarded to those who maliciously misuse the system. Similarly, cancer-like programs in the system - those programs which go awry from their intended functions - can often be detected by algorithms programmed into the system. These algorithms might include such methods of repeated cycle detection, etc.

The duplication of the actions of the immune system is almost certainly not in place in any system operating today. The vastness of the variety of programs that would need to be present would make such a system impossible in its own right, in addition, establishing a method of marking programs as legal without detection by pirates and others wishing to break the system would be difficult, as any coding could be detected and broken by anyone with the resources to do so. At best, a coding scheme related to the time a program was entered into the system could be devised, although it is difficult to imagine that such a scheme could not be thwarted.

5.4.3 Applicability of ecological theory on security.

Although ecological literature may be of little value in providing ideas on the protection of computer systems from external sources of corruption, some ideas can be found for protection of individual files. Some general ideas on cost-effectiveness and possible methods of reducing internal corruption sources will be given.

Camouflage is a method whereby an organism protects itself from predation by looking like something other then itself. Camouflage comes in two forms, mimicry and cryptic coloration and behaviors.

Cryptic colorations and behaviors in general involve organisms which avoid predation by looking like some inanimate object. Many organisms utilizing this form of protection look like stones or bird droppings or other inanimate portions of their environment. Other organisms have patterns which allow them to blend in with their backgrounds and behaviors which cause them to situate themselves in such a manner that they do blend in.

Another form of crypsis involves disruptive coloration which breaks up the pattern of the organisms outline in an attempt to prevent detection and identification of the organism apart from its background.

Mimicry is a technique of avoiding predation by looking like another organism which is inedible for one reason or another. Two types of mimicry exist, Batesian mimicry and Mullerian mimicry. Often these mimics mimic a bold coloration pattern, called aposematic coloration, which is designed to make the owner more visible to potential predators. Organisms with aposematic coloration "want" the predator to spot and identify them so that they are not mistaken for an edible organism and killed. Such organisms always have something about them that makes them inedible or extremely unpleasant to eat. The aspect which makes them unpalatable to predators may include toxins or dangerous physical structures. Examples of such organisms include coral snakes, certain toads with poisonous skins, and Monarch butterflies and other butterflies which feed on toxic plants and store the toxins in their own bodies.

Batesian mimics are organisms which look and/or act like toxic organisms which try to advertise their presence. Viceroy Butterflies are a well known example of Batesian mimics and are mimics of Monarch Butterflies. Organisms which rely on Batesian mimicry usually are lower in population and emerge slightly later than the species of which they are mimics. By the time the mimics emerge, any inexperienced predators have probably already tried to eat the host species and have learned to avoid the color and/or behavior pattern it displays. By keeping at a lower population than the host, and by emerging later, the mimics avoid the possibility that predators will eat several of them before it learns that the color and/or behavior pattern is one to be avoided.

Mullerian mimics are several related species which possess aposematic coloration and mimic each other, thereby increasing the probability that any predator they encounter will be educated. Batesian mimics on such complexes of species can successfully increase their populations without adversely affecting the success of their defense.

5.4.3.1 Use of Hiding/Camouflage Techniques for IDOs.

Encryption would not parallel the use of camouflage as an encrypted file is usually a file in need of protection. Utilization of biological cryptic techniques would probably involve camouflaging a sensitive file as a more common nonsensitive file. This could involve using some encoding technique which makes the encryption appear to be a normal file of a "worthless" type to system invaders. Taking this technique a step further, one might break the file contents up among several "normal" forms - a technique not easily available to organisms, but may be represented by social insect colonies.

To break up the outline of a file, streams of data used to transfer files over communication lines might be altered to disguise the head and tail of any given file, thus making the task of finding the "right" file more difficult. In addition, a sensitive file might be broken up by some algorithm and sent over several different communication devices or at varying times over the same communication device, thereby reducing the content being sent over the device at any one time to a meaningless string of symbols or bits. If the latter form of such a technique were used, it would probably be necessary to send at least the timing or fracturing algorithm over a separate medium to avoid its use by a wiretapper.

5.4.3.2 Increasing cost to decrease predation in biological systems

One of the methods used by organisms employing aposematic coloration as addressed above is the storage of toxins within the organisms body. This makes the organism potentially dangerous for a predator to eat. The Monarch Butterfly mentioned above is an organism which utilizes this technique by storing toxins produced by milkweeds.

A method used by many organisms is to make themselves too expensive for effective use of their resources by a predator. Trees store resins and other waste materials in their trunks to make them undigestible by herbivores. Some plants stores minerals or compounds which wear the teeth of herbivores or poison the herbivore. There are animals with hard shells, scales, or quills which make them costly to kill and eat.

5.4.3.3 Increasing the Cost of Data Theft in Computer Systems.

A defense in computer systems which might parallel the storage of toxins would be to place incorrect and potentially damaging data in files sensitive to corruption. A method of increasing the cost of obtaining sensitive data could include the storage of large quantities of meaningless data in sensitive files to make it difficult to discover which data is in fact the important data. Encryption is another example of this technique.

5.4.3.4 No security as security.

Perhaps the widest form of security used in biological organisms is none at all. Security is almost always costly to those organisms that use some method to secure it. In some situations, that expenditure might be better used to directly accomplish the goals of the organism. An additional facet to this approach to security is that the organism is exposed to danger for a shorter period of time. The reduction of the time necessary for the organism to accomplish its immediate goals may be sufficient to make the organism not worth the cost of pursuing. Evidence in support of the claim that this type of security works is the fact that 840,000 of 861,510 species on earth are less than an inch in length, and 240,000 of these are also less than 0.01 inches [Moro82] - which doesn't leave much room for security measures.

5.5 Summary

In this chapter several relationships between biological systems and Intelligent Data Object Management Systems have been discussed. One of the relationships discussed involved the application of ecological ideas and terminology to IDOMSs. Another discussed the application of optimal foraging ideas as developed for organisms to IDOs trying to optimize their procurement of data and authorizations in IDOMSs. Still another area covered involved the application of sociobiological ideas to the relationship between instruction sets and the group dynamics of IDOs. Finally, the security of biological systems was discussed both at the organismic and the ecosystem level. The possible application of these security measures to IDOs was presented. Although each of these areas of discussion could be considerably extended, and numerous other areas of application of ecological and/or other biological principles exist, the discussion in this chapter should demonstrate the utility of further exploration of the relationship between IDOMSs and biological systems.

A summary of work presented in this thesis will be given in Chapter 6. In addition, suggestions will be made as to possible areas for further research on the relationship between IDOMSs and biological systems, along with possible applications of work already completed.

Chapter Six

Results and Suggested Areas for Further Study

6.1 Results

In this thesis I have demonstrated a relationship between Intelligent Data Objects and biological organisms (see table 6.1), and have demonstrated a similar relationship between the environments in which they operate (see table 6.2). This relationship has been formalized in a model which which we call an abstract intelligent entity (see figure 6.1). The components of this abstract intelligent entity are then compared to the corresponding components of both biological organisms and IDOs. A model of an intelligent entity based system is presented and compared to ecological systems and intelligent data object systems.

Organisms operating in ecological systems and IDOs operating in IDOM IDOMSs have been shown to be implementations of the same model. It is therefore reasonable to use one system to model the behavior of the other. Care must be taken, however, to avoid extending comparisons beyond the limits of the model. Biological organism implementations of the model have been used to model implementations of IDOs and their behaviors when operating in an IDOMS. Use of the model in this fashion can allow more efficient construction of a prototype IDOMS. Once an IDOMS is in existence, the model can be used to

Table 6.1 - A comparison of IDO's and Biological Organisms.

Components	IDO	Biological Organism
Instruction Set	A set of programming language-like instructions	Chromosomes, "hardwired reflex arcs, learned behaviors
Environmental Senses	Ability to detect certain machine states	5 basic senses + other less well known senses
Ability to store matter and energy	Data structures for storage of data	Chemical storage of matter and energy
Basic structure	Physical structure defined in data structures, etc.	Physical structure based on atoms, molecules, and physical forces.
Ability to process and store information from the environment.	Can obtain and store information from the environment	Higher animals can learn new behavior patterns.

justify comparisons which might help to improve the IDOMS. Some behavioral comparisons have been made in this thesis (summarized in table 6.3).

6.2 Suggested Areas for Further Study

Areas for future study include three main areas of effort. First, the model should be tested with an implementation of an IDOMS. If the model is shown to be applicable, it should be extended to more completely represent the relationship between IDOMSs and ecological

Table 6.2 - A comparison of IDOMS's and Ecological Systems.

Components	IDOMS	Ecological System
Other entities	IDOs.	Organisms.
Resources	Data, authorizations, and CPU time.	Matter and Energy
Time	Important factor.	Important factor.
Space	Memory, Important factor.	Physical space, Important factor.
Structure of environment	Complex.	Complex.

systems, and the relationship between IDOs and biological organisms. In addition, the current model could be used to justify further exploration of possible applications of ecological principles to IDOs and IDOMSs. Further explanations of each of these topics is given below.

6.2.1 Verification of the model.

The next step in the development and use of this model should include the building of a prototype IDOMS to test the correctness and applicability of the model. As a part of this testing process, the behavioral predictions from chapter five can be tested.

The ability to take up space and possess structure.
Occupying space is a requirement for existence in entities
Some structure is necessary to hold information and to allow actions

An instruction set which allows it to react to the environment in a manner which will potentially optimize the fulfillment of its goals.
An instruction set allows the entity to react to its environment in a non-random manner.

The ability to act on its instruction set.
An instruction set must be tailored to the entity which is to carry out the instructions, otherwise it is useless.

The ability to obtain selected information about its environment.
Environmental information is required if the entity is to react to environmental states.

A set of goals which include obtaining a set of resources within a given amount of time.
Without goals the entity has no purpose. In such a case an instruction set is useless since the entity has no reason to execute the instructions.

Figure 6.1 The components of an intelligent abstract entity.

6.2.2 Extension of the model.

Once the model has been demonstrated as a reasonable representation of both ecological systems and IDOMSs, further extensions of the model should be made.

One area which should be examined is the relationship of large system theory to the current model. Large system theory attempts to discover unifying concepts of large systems. Apparently, much of this

Table 6.3 - Selected Ecological Principles applied to Intelligent Data Objects.

Area of discussion	Ideas discussed.
Interspecific Interactions	Terminology and ideas directly applicable.
Optimal Foraging	Prey choice, foraging for patchy resources, optimal return times to renewable patches
Sociobiology	Hierarchical goal sets, applicability of group selection arguments, applicability of relatedness to IDOs
Security	Absolute security impossible in ecological systems, Immunological systems in organisms may provide helpful information, discussion of applicable information from ecological systems

theory discusses aspects of systems which are less detailed than those covered in this thesis. Although no information directly applicable to this thesis has been found, information may exist which will increase understanding of IDO systems through relation of IDOMSs to large systems in general.

Another area of research which may prove fruitful is the modeling of functionality of instruction sets both on future implementations and on other IDOMSs operating in the current system. If it can be shown that the same functionality exists in both systems it may be reasonable to apply relatedness and kin selection principles to a greater degree than that discussed in chapter 5.

6.2.3 Behavioral modeling of IDOs.

In addition to extensions of the current model, the model can be used to further model optimal behavior of IDOs. Several areas of ecological and biological literature are discussed in section 6.3 which may prove fruitful areas of information.

6.2.4 Formalization of the model.

Finally, more detailed formal models of both the abstract intelligent entity model and the intelligent entity based system should be made. Such models would allow more exact predictions of the system behavior, which would then allow more accurate defining of IDO models.

6.3 Major Areas of Ecological and related Biological Literature

There are three major areas of literature with ecology and related areas of biology which hold promise of providing usable theory for application to IDOs and IDOMSs. These areas are behavioral ecology, population ecology, and sociobiology.

Behavioral Ecology provides an examination of behaviors which allow organisms to interact with their environment in an optimal fashion. Many behaviors exhibited by organisms might be reasonably applied to IDOs operating in IDOMSs to optimize their actions as they seek to fulfill their goals.

Population Ecology is the section of ecology which deals with describing populations in terms of size, structure, interactions, changes in population over time, etc. Techniques useful in describing populations of organisms such that they can be studied in a scientific manner should also prove useful in studies of population statistics and behaviors of IDOs of similar as well as differing types.

Sociobiology is an area of biology which examines the "programming" of organisms (mostly animals) which act in a manner which may have the immediate consequence of reducing their chances of fulfilling their immediate goals but have the long term consequence of, on the average, increasing the fitness of the individual by increasing the survival of their genes.

This area of biology may provide further insight into the wide ranging effects of instructions. This literature can serve as a possibly different view point on interactions and their immediate and long-term effects. An extension of the model should be made concerning functionality of the relationships among instruction sets before attempting to apply sociobiological principles.

This has applicability in programming of complex, concurrent, computer systems in that individual forms (IDOs) which require processing are almost invariably only a small part of the work load of a company, i.e. a group working toward common goals. Thus, if a form is programmed to increase the likelihood that it will accomplish its (narrow) stated goal, without regard for how this affects the processing tasks of the company as a whole, the form may ultimately succeed in its own efforts and in bringing down the company (worst case scenario).

SELECTED BIBLIOGRAPHY

[Alco75] Alcock, John., Animal Behavior: An Evolutionary Approach. Sinauer Associates, Inc., Sunderland, Massachusetts. 1975.

[Bara77] Barash, David P., Sociobiology and Behavior, Elsevier North-Holland, Inc., New York, NY. 1977.

[Baye79] Bayer, R., Graham, R.M., and Seegmuller G. Operating Systems: An Advanced Course. Springer-Verlag, New York. 1979.

[Bego81] Begon, Michael and Mortimer, Martin., Population Ecology, Sinauer Associates, Inc., Sunderland, Massachusetts. 1981.

[Bish86] Bishop, Sandra K., Access Rights for Intelligent Data Objects. Master's Report, Kansas State University, Manhattan, Kansas. 1986.

[Busa85] Busack, Nancy L., The Intelligent Data Object and its Data Base Interface. Master's Report, Kansas State University, Manhattan, Kansas. 1985.

[Char76] Charnov, E. L., Orians, G. H., and Hyatt, K. The ecological implications of resource depression., American Naturalist, 110:247-59. 1976.

[Cody75] Cody, Martin L. and Diamond, Jared M., editors, Ecology and Evolution of Communities., The Belknap Press of Harvard University Press, Cambridge, Massachusetts. 1975.

[Cox86] Cox, Brad, Object Oriented Programming: An Evolutionary Approach. Addison-Wesley. 1986.

[Curt83] Curtis, Helena., Biology, Worth Publishers, Inc., New York. 1983.

[Davi76] Davies, N. B. Food, flocking and territorial behaviour of the pied wagtail (Motacilla alba yarelli) in winter, Journal of Animal Ecology, 45:235-54. 1976.

[Davi77] Davies, N. B. Prey selection and search strategy of the spotted flycatcher (Muscicapa stiata), a field study on optimal foraging. Animal Behaviour, 25:1016-33. 1977

[Davi81] Davies, N. B. and Houston, A. E. Owners and satellites: the economics of territory defence in the pied wagtail Motacilla alba, Journal of Animal Ecology, 50:157-80. 1981.

[Deit84] Deitel, Harvey M., Operating Systems. Addison Wesley Publishing Company, Reading, Massachusetts. 1984.

[Elne78] Elner, R. W. and Hughes, R. N. Energy maximisation in the diet of the Shore Crab, Carcinus maenas (L.)., Journal of Animal Ecology, Vol 47, pp. 103-116. 1978.

[Gamo73] Gamov, R. I., and J. F. Harris. The infrared receptors of snakes., Scientific American, 228:94-100. 1973.

[Gant85] Gantt, D. M. Management of an Intelligent Data Object, Masters Report, Kansas State University, Manhattan, Kansas 1985.

[Geha82] Gehani, N. H., The Potential of Forms in Office Automation., IEEE Transactions on Communication. Vol. Com-30, No. 1, January, 1982.

[Gill75] Gill, F. B. and Wolf, L. L. Economics of feeding territory in the golden-winged sunbird. Ecology, 56:333-45. 1975.

[Gord72] Gordon, M. Animal Physiology: Principles and Adaptations., Macmillan, New York. 1972.

[Hick82] Hickman, Cleveland P., Roberts, Larry S., and Hickman, Frances M., Biology of Animals, The C. V. Mosby Company, St. Louis. 1982.

[Hick74] Hickman, Cleveland P., Sr., Hickman, Cleveland P., Jr., and Hickman, Frances M., Integrated Principles of Zoology, The C. V. Mosby Company, St. Louis. 1974.

[Hone87] Honeyman, Janice C., A Form Definition Language for an Intelligent Form. Doctor's Dissertation, Kansas State University, Manhattan, Kansas. 1987.

[Huml86] Huml, Kathy Petersen, Intelligent Data Object Management System (IDOMS). Master's Report, Kansas State University, Manhattan, Kansas. 1986.

[Kami79] Kamil, A. C. Systematic foraging for nectar by Amakihi (Loxops virens). Journal of Comparative Physiological Psychology, 92:288-396. 1979.

[Keet73] Keeton, William T., Elements of Biological Science, W.W. Norton & Company, Inc., New York. 1973.

[Kisl76] Kislalioglu, M. and Gibson, R. N., Prey 'handling time' and its importance in food selection by the 15-spined stickleback Spinachia spinachia (L.)., Journal of Experimental Marine Biology and Ecology., Vol 25, pp. 151-158. 1976.

[Kitc83] Kitching, R.L., Systems Ecology: An Introduction to Ecological Modelling. University of Queensland Press, St. Lucia, Queensland, Australia. 1983.

[Klop67] Klopfer, Peter H., and Hailman, Jack P., An Introduction to Animal Behavior: Ethology's First Century. Prentice-Hall, Inc., Englewood Cliffs, New Jersey. 1967.

[Kreb81] Krebs, John R., and Davies, N.B. An Introduction to Behavioural Ecology. Sinauer Associates, Inc., Sunderland, Massachusetts. 1981.

[Kreb78] Krebs, John R., and Davies, N.B. Behavioural Ecology. Blackwell Scientific Publications. Oxford, England. 1978.

[Kroe83] Kroenke, David. Database processing. Science Research Associates, Inc. 1983.

[Lee86] Lee, Kum-yu Enid, Privacy and Security of an Intelligent Office Form. Master's Report, Kansas State University, Manhattan, Kansas. 1986.

[Lind77] Linday, Peter H., and Norman, Donald A., Human Information Processing: An Introduction to Psychology. Academic Press, London. 1977.

[MacA66] MacArthur, Robert H., and Pianka, Eric R., On optimal use of a Patchy Environment, The American Naturalist, Vol 100:603-610. 1966.

[Malo86] Maloy, Thomas P., An Intelligent Message System. Master's Report, Kansas State University, Manhattan, Kansas. 1986.

[Marg68] Margalef, Ramon, Perspectives in Ecological Theory. The University of Chicago Press, Chicago, London. 1968.

[May81] May, Robert M., editor, Theoretical Ecology - Principles and Applications. Sinauer Associates, Inc., Sunderland, Massachusetts. 1981.

[McBr83] McBride, R. A. and Unger, E. A. Modeling Jobs in a Distributed System. ACM, 1983.

[McBr87] McBride, R. A., Personal Communication. 1987.

[Moro82] Morowitz, Harold, Beetles, ecologists, and flies. Science 82, pp. 28 - 30. 1982.

[Odum83] Odum, Eugene P., Basic Ecology. Saunders College Publishing. 1983.

[Odum82] Odum, Howard T., Systems Ecology: An Introduction. John Wiley & Sons, New York. 1982.

[Oria69] Orians, Gordon H., The Study of Life: An Introduction to Biology. Allyn and Bacon, Inc. Boston. 1969.

[Orr76] Orr, Robert Thomas, Vertebrate Biology, W. B. Saunders Company, Philadelphia, PA. 1976.

[Pyke77] Optimal Foraging: A Selective Review of Theory and Tests, The Quarterly Review of Biology, Vol. 52:37-55. 1977.

[Putm84] Putman, R. J., and Wratten, S. D., Principles of Ecology. University of California Press, Berkley and Los Angeles, California. 1984.

[Ryko86] Rykowski, Ronna W., Design of the IDO for the Intelligent Data Object Management System (IDOMS) Project. Master's Report, Kansas State University, Manhattan, Kansas. 1986.

[Shoc82] Shoch, John F., and Hupp, Jon A. The "Worm" Programs-Early Experience with a Distributed Computation. Communications of the ACM, Vol 25:3, pp. 172-180. 1982.

[Sewc86] Sewczwicz, Richard P., Form Definition Language for Intelligent Data Objects. Master's Report, Kansas State University, Manhattan, Kansas. 1986.

[Silj78] Siljak, Dragoslav D., Large Scale Dynamic Systems: Stability and Structure. North-Holland, New York, Amsterdam, Oxford. 1978.

[Smit84] Smith, C. C., and Reichman, O. J., The Evolution of Food Caching by Birds and Mammals, Annual Review of Ecology and Systematics, Vol 15:329-51. 1984.

[Tane81] Tanenbaum, Andrew S., Computer Networks. Prentice-Hall, Inc., Englewood Cliffs, N.J. 1981.

[Tsic85] Tsichritzis, D. C., ed., Office Automation. Springer-Verlag, New York. 1985.

[Ullm82] Ullman, Jeffrey D., Principles of Database Systems. Computer Science Press, Rockville, Maryland. 1982.

[Unge87] Unger, Elizabeth A., Personal Communication. 1987.

[Welt82] Welty, Joel C., The Life of Birds. Saunders College Publishing, New York, NY. 1982.

[Wils75] Wilson, Edward O., Sociobiology: The New Synthesis. The Belknap Press of Harvard University Press, Cambridge Massachusetts, and London, England. 1975.

[Wynn62] Wyne-Edwards, V. C., Animal Dispersion in Relation to Social Behaviour. Hafner, New York. 1962.

A MODEL OF AN ABSTRACT INTELLIGENT ENTITY AS A MEANS OF COMPARISON OF INTELLIGENT DATA OBJECTS (IDO) AND THEIR MANAGEMENT SYSTEM (IDOMS) TO BIOLOGICAL ORGANISMS AND ECOLOGICAL SYSTEMS

by

STEVEN C. HANSEN

B. S., St. John's University, 1978
M. S., Kansas State University, 1983

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1987

ABSTRACT

This thesis presents a model of an Intelligent Entity as an abstraction of the essential qualities of both an Intelligent Data Object (IDO) and a biological organism. Also presented is a model of the system within which an Intelligent Entity operates. This system is an abstraction of both an Intelligent Data Object Management System and an Ecological System. To develop the model, a description of Intelligent Data Objects and their Management System is given, followed by a description of biological organisms and Ecological Systems. The model is then presented and each component is shown to be representative of both of the systems on which the model is based. Ecological principles are used to predict relationships which should exist in Intelligent Data Object Management Systems.