INFORMATION DYNAMICS: AN INFORMATION-CENTRIC APPROACH TO SYSTEM DESIGN

Ashok K. Agrawala
Department of Computer Science
University of Maryland
College Park, Maryland 20742
agrawala@cs.umd.edu

Ronald L. Larsen
Maryland Applied Information Technology Initiative
University of Maryland
College Park, Maryland 20742
rlarsen@deans.umd.edu

Douglas Szajda
Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland 20742
szajda@umiacs.umd.edu

KEYWORDS: system design, integration, analysis, information

ABSTRACT

Acquisition, distribution, management, and analysis of information are the fundamental purposes behind most complex constructed systems and infrastructures, and yet a process-centric approach is fundamental to the design and implementation of such systems. Since *information* is the essential commodity in these endeavors, we believe that an effective design should take into account the fundamental properties of information: its characteristics, its representation, its value, its temporal dynamics, its fusion, its distillation, etc. Information Dynamics is an attempt to bring a degree of rigor to the understanding of the nature of information itself and how it is used in the pursuit of system objectives.

INTRODUCTION

The most significant and far reaching technological development of the 2nd half of the last century is the development of information technology (IT). Today, IT touches nearly every aspect of our lives. And the rapid growth of new applications and new technologies assures that the trends of the 90's will accelerate in the current decade. The hardware technology exists to create ever more complex systems. However, our ability to design, implement, operate, maintain and support complex systems lags. We believe that one reason is the paradigm used in system design, namely that of the *process-centric* view of system design.

We propose Information Dynamics as an alternate paradigm in which we take an *information-centric* view. In this approach we explicitly consider the role information plays in a system, and design the system taking into account what information is needed and when, who has the information, and what happens to the information as it moves from one place to another. In an Information Dynamics framework, information is treated as a dynamic entity and its dynamics (e.g., location, currency, value) are explicitly considered. Any processing of information is referred to as "action" that is carried out under the control of "choice". Any action that carries out a transformation or related processing of information consumes resources, therefore requiring resources for some period of time, i.e., occupying some subspace in resource/time space. All actions take time and, therefore, have an impact on the dynamics of information that has to be considered in the design of a system. "Choice" defines the control mechanisms for actions in this framework. Further, we associate a "value" (sometimes called "utility") for the information in a particular "Context" and recognize that the value of information typically changes with time within a given context.

The use of information for effecting control and other decision processes is, of course, not new. Physical systems respond adaptively to linear-quadratic-gaussian (LQG) controllers, for example, when the physics is well understood and controllers have rigidly compartmentalized responsibilities. But decision making in network-based, distributed systems for which there is no nice physics poses a different class of problem. Early investigations into distributed decision making lead to a wealth of research in game theory and later team theory (e.g. Bascar and Olsder 1982; Greenwald 1998; Owen 1968), but none of this work explicitly considered the temporal effects on the value of information, and how that affected system performance. Recent literature on autonomous agents (e.g. Stone and Veloso 1998; Washington 1998) reports work in distributed, multiagent decision making (using, for example, state space approaches such as Markov Decision Processes) but still fails to consider the temporal issues underlying the usage and value of information. So, while information, as an entity, is used in many disciplines, and differently in most of them, we are not aware of any approaches that explicitly approach the problem with a temporal perspective. We believe that Information Dynamics is not an incremental extension to any existing work, but, informed by this work, takes an orthogonal look at the problem.

In the following sections, we discuss aspects of the nature of information, its dynamic properties, and some of the implications for complex system design. We follow with an example of the theory applied to a networking problem.

WHAT IS INFORMATION?

We all have an intuitive notion of what information is, but making this precise is hardly intuitive. Information is a property, characteristic, or description of something physical, logical, virtual, or conceptual. That "something" may be other information. It may be a group, an action, a choice. Or it may be a relationship between any of these things.

Information cannot exist in isolation and has no value without a context. It refers to some entity that can be logical or physical, can be another piece of information, can be information about information, etc. Relationships between information may be direct or indirect, and exist whether they are enumerated or not. Relationships may be static or dynamic.

The causality principle applies. Information in the present can only affect the future; it cannot change the past. It may change the interpretation of the past, but it cannot change the past, itself. Further, delays involved in the movement of information assure that our knowledge of a remote entity is necessarily delayed; the "present" may (and typically will) actually reflect a system's state at some past time instant.

As any system consists of a number of components, considering all possible pieces of relevant information and their relationships yields an arbitrarily large amount of information. In a typical system design only a small amount of relevant information is collected and used.

Note that when two pieces of information are related, their relationship may be considered a higher level of information. Processing establishes the validity of a relationship, or it may be used to derive one piece of information from the other. When the *relationship* is explicit, one piece of *explicit* information yields the second piece *implicitly*. In this regard, we consider information that can be derived from other information based on known relationships as implicit information. Note that deriving such implicit information, i.e. making it explicit, requires some processing for carrying out the interrelationship calculations.

Information Has Value

Every piece of information can have value *within a given context*. The value of information depends on its use and/or purpose. This is the role of context. Clearly the value of information changes with time and depends on the context and the frame of reference. The value can, and often will, depend on its relationships to other pieces of information. Within a context, information value can typically be quantified.

Value Of Information Is Time Dependent

The value of information typically decreases with time within a context. If the information is static, its value may also be static (but will change with context). The value of a piece of information may increase or decrease due to the discovery or instantiation of new relationships, or due to the transformation of some information from implicit to explicit.

Information Variable

We use the term "information variable" to refer to a piece of information and its associated metadata. An information variable consists (at least) of the following:

- Descriptor: Qualitatively defines the information variable and supplies the rules for interpreting its "magnitude".
- Type: "primitive" or "composite". Here primitive refers to an information variable as a basic element of information whereas composite refers to an interrelationship.
- Magnitude: a bit string associated with the information variable (traditionally referred to as its value, but that term is used in another way here). Not all information is quantifiable. For such information, the bit string may represent a text string, or any other symbolic form. The Descriptor provides the rules for interpreting the string.
- Provenance: a time indicator, showing the time when this information was collected or acquired or the time it refers to. It may also include a location indicator showing the location the information refers to or the location from which it was acquired.
- Confidence Indicator: defines the confidence measure in the magnitude. This measure is an indicator of the quality of the magnitude without any regard for the use that may be made of this information.
- Context vector: defines the context within which the information is relevant, enabling a relevance computation (e.g., cosine or dot product) to establish the importance of this information variable to the task at hand. The importance indicator may be only a subjective measure of the importance of the information.

The confidence indicator and context vector are both required in order to compute the value of information for this variable. The confidence indicator does not change with time but the context vector may need to accommodate the age of information or the passage of time.

For an information variable we can define two additional functions. The first is a time function that indicates how the context vector changes over time. This may be a probabilistic function. The second is a fusion function, which defines how two pieces of information can be combined, or "fused."

Note that while all of these components are required to fully define an information variable, many times only some of these components may be known or available. Others are either left unspecified or undefined and, hence, unused.

REPRESENTATION OF INFORMATION

We have used the term information to refer to abstract, or ideal, information (e.g., the concept behind an integer two). In order to carry out any manipulations on information it is essential to have a representation for it (e.g., a decimal symbol "2" or binary symbol "10"). Such representations are essential to store, move, or process the information. We note that some representations may use implicit information. For example when we use a data structure, it not only defines the representation of some quantities but also some relationships.

An interesting question arises regarding the representation of implicit information. As the implicit information is expressed through the interrelationships that can be processed to make the implicit information explicit, we may consider the known relationships as the implicit information. However, for the implicit information to convey the same meaning to the sender and the receiver, the two sites have to have the same understanding of the relationships and have a common understanding of how to extract any implicit information (make it explicit) from some given explicit information. They must agree on a common *context*.

Capture Of Information

We recognize two processes for capturing information: direct observation and inference. Direct observation may be through sensing/monitoring/measurement, etc. Such observations may be made directly or indirectly. Inference typically involves the use of interrelationships to make implicit information explicit. In this process we may use induction as well as deduction to capture new information.

It is interesting to note that in this context we may treat mathematics as a framework of interrelationships with a precise description of context and applicability. A framework of deduction and induction is also specified and we use these relationships in a variety of contexts invoking the implicit information. Analytical results are nothing but specified interrelationships with a description of their applicability.

Storage Of Information

In order to store information we need a *representation* for it. To use it, we need to retrieve it. A representation suitable for storage may not contain all the components or all the interrelationships. On retrieving we may be able to calculate some of them while others may be lost forever. In particular, information relating to time can be lost unless time stamping is explicitly done. Note that storing information is an action; therefore it requires resources and takes time. And all actions generate additional information that may or may not be captured.

Movement Of information

As with the storage of information, only explicit information can be moved from one location to the other. Further, the representation used for moving the information has to be such that the receiver can interpret it correctly. This requires a common understanding between the sender and the receiver which, in turn, may require conventions, protocols etc.

Moving information is an action; it consumes resources and time. Systems use networks to move information. Of course, a number of issues have to be resolved in moving information. Consider an infrastructure that can move information from location x to location y. First we have to decide who initiates the movement of information and why. How does Y know that X has the information it needs and how does X know that Y needs that information? The knowledge about who has what information is a crucial part of the design of a distributed system. This phenomenon is quite evident on the Web today.

Let us consider some immediate implications of information movement. Moving information from location x to location y takes t_{xy} time. As a direct consequence of this, at y we can only get information from x that is at least t_{xy} old. Therefore, in a distributed system it is impossible to capture the current state of the entire system at any one location!

Value Of information: Confidence Indicator And Context Vector

We may associate a value attribute to any information using a confidence indicator and a context vector. Clearly the value of information depends on its use or purpose (what we call context). The value of information changes with time, typically decreasing with time. When the underlying system is static, the value may, likewise, remain static. Under some circumstances the value may increase with time, as later information makes it more valuable. In this regard the value of information may also be associated with the interrelationships.

The confidence indicator may be represented by uncertainty models. For example, we may be interested in the waiting time at a router. When we measure it at time t, the measured value may be very precise. However, without having any additional measurements, the estimate of the waiting time will change, and in particular its variance will increase, moving towards the steady state value and the steady state variance.

Information Fusion

In the natural world, the amount of available information monotonically increases (barring catastrophic events such as the burning of the Alexandria library). Its effective management requires techniques for reducing, or aggregating, it while maintaining its quality. One way of distilling the captured information is by retaining higher-level information reflected through relationships.

Fusion functions, one of the key elements of information dynamics, are another method for distilling information. The fusion function permits us to define methods by which multiple views of a single information variable are combined. Aspects of fusion include specifying the effects on magnitudes, values, and confidence indicators. We envision several classes of fusion functions, corresponding to the characteristics of the information under consideration. For example, if we have multiple measurements of a single variable, we may fuse these measurements using minimum variance estimates to determine a single magnitude. If, on the other hand, we have single measurements of multiple variables, we may call upon known relationships among the variables to establish a statistically valid fusion.

Using Information

The use of information requires *action*. Action can create or capture information, store it, move it, transform it, or destroy it. Information can be processed to make implicit information explicit, to initiate another *action*, i.e. defining a “choice”, or to activate a physical operation as an output.

As we use it here, the term *action* refers to processing that consumes resources and takes time. As resources reside at specific locations, actions are carried out at *locations*. It typically uses information as input, and starts under the control of choice. The outcome of an action may be additional information, choice, storage of information, movement of information, or some physical results in the form of commands to actuators, etc.

Choice

We use the term choice to define the control function. Choice defines what action has to be carried out where, at what time, under what conditions, and using what resources. It is based on the information and its location/time or value.

Note that choice is generated by processing information. Therefore the relationship between *information* and *choice* must be part of the implicit information. The interrelationships between information and choice may be fixed, leading to a hardwired design as a design-time choice. When the relationships are dynamic the choice has to reflect it.

Distributed Systems

Consider the implications of information dynamics on a distributed system. Such a system has a collection of entities (processing resources) capable of carrying out certain operations. A specific distributed system, designed to carry out a specific mission, uses physical resources to carry out actions and to store and move information. When such a system is interacting with an external physical system it also has sensors and actuators.

A distributed system may be considered a collection of nodes with a defined network infrastructure for communication among them. Let us examine the participation of a node in processing. The node maintains its view of the universe in the form of “perceived reality” which is based on:

- Prior model of the Universe,
- Explicit information received and processed

The explicit information is processed to integrate it with the perceived reality and is based on the model of the universe. Depending on the model, which is nothing but a collection of interrelationships, it may permit the new information to change the model.

At any node in a distributed system, all actions are initiated using the knowledge of its perceived reality that is not always explicitly defined or represented. The explicit representations may only have been used at design time, and the final system may only contain those parts that are considered essential for operation, retaining only such relationships that may be activated at runtime.

A far-reaching consequence of the movement of information is that **the perceived reality at any node CANNOT be assured to be the same as the actual reality at any remote node.** Transmission delays assure that information received from any remote node is, by definition, historic. Further, it is not sufficient to receive messages; they must be interpreted and processed to integrate them with the local perceived reality. While the perceived reality of a node cannot be assured to be the same as the actual reality of a remote node, it can be consistent with models of remote reality. (Lamport

(Lamport 1978) considered similar issues in the context of message ordering in distributed systems. He reached a "parallel" conclusion---that when transmission delays are not negligible compared to the time between events, it is sometimes impossible to say that one event occurred before another.)

Receiving Information

Message transfer is the primary means of communication in a distributed system. Upon receiving a message, a node processes it syntactically to interpret its structure. Then it processes it semantically to interpret its content, converting it into explicit information. The explicit information is then integrated with perceived reality, potentially altering the current state and modifying the choices made for processing of implicit information.

KEYS TO INFORMATION DYNAMICS

In Information Dynamics we take an information-centric view of a system and explicitly take into account the time dependent aspect of information, the value of information, and the role of implicit information. Within this framework we organize and design a system based on the desired dynamics of information, taking into account the constraints of the dynamics of information.

We believe that Information Dynamics provides a new and vital framework for the design and implementation of complex systems. It is also useful for planning and decision making. Effective decisions must be based on the appropriate perceived reality, taking into account the model of the universe and its dynamics, along with the sources of information and their role in defining dynamic choices. Clearly the value of information in terms of confidence indicators and context vectors, along with the changes in the value of information with time, play a key role in the planning process--they reflect the dynamics of information.

EXAMPLE: ROUTING

As a concrete example of the impact of information dynamics on a practical problem, consider the link state routing in a computer network. In this method, routes are chosen to be the shortest path from the source to the destination as determined according to the current knowledge of the state of links. The common practice is for the links to periodically measure their state, determine the waiting time, and broadcast this information so that all nodes have it and can use it in route determination.

Let us now consider the basic characteristics of the performance of a link. In a typical network the link is continuously transferring packets, handling the load as it is presented to the link. If we consider this link as a server, we can characterize its steady state behavior in terms of the mean, w, and the variance, v, of the waiting time. Let w(t) be the waiting time at this particular link as measured at the link at time t. Assuming that the measurement is done correctly, the variance of this measurement, v(t), is zero. Given no additional information about the state of the link, our estimate of the waiting time $w(t_1)$ at some later time t_1 will have to be based on w(t) and our knowledge of w. This estimate will have a variance, $v(t_1)$, which will not be zero. In fact, the variance will be an increasing function of the difference t_1-t, tending towards the steady state value v. Given w(t) and v(t), the actual values of $w(t_1)$ and $v(t_1)$ can be estimated with knowledge of the stochastic behavior of the link.

In this example, the basic information variable is the waiting time estimate for the link and the variance estimate is its confidence indicator. Recognizing that the transfer of the magnitude of w(t) to any other nodes in the network takes time, we require that any new estimates be made taking into account the information dynamics of the situation. Depending on the characteristics of the link, the estimates $w(t_1)$ and $v(t_1)$ may come so close to the steady state values w and v that the new measurements will have no significant impact on the link information retained by another node. As a consequence, we can significantly reduce the communication required for supporting link-state routing while at least maintaining the quality of routing decisions, if not improve them, by taking into account the variance of the delay estimates. Each node does need the steady-state information about the links. Note that if the steady state conditions change regularly, that knowledge can also be given to individual nodes. The key impact, though, is that by taking the value of information into account, the information dynamics approach allows us to improve the design of the routing scheme.

It is important to understand that although the above example uses statistical measures, the information dynamics framework is not limited to handling quantitative information. We believe that it can be equally effective in using other forms of information including qualitative information and its value expressions.

CONCLUSIONS

The management of information is at the core of many complex systems and yet we employ a process-centric approach to the design and implementation of such systems. Since information is the essential commodity in these endeavors an efficient design should take into account the fundamental properties of information. Information Dynamics is an attempt to bring a degree of rigor to the understanding of the nature of information itself.

The first principle in understanding the generic nature of information is recognizing the distinction between information and its representation. Computer systems are only capable of manipulating representations and it is through the processing of representations that we attempt to carry out the processing of information. These representations are limited in that they capture only a very limited portion of the generic information. Moreover, the various processing steps change the nature of information in ways that are not necessarily intended or anticipated. We claim that *implicit information* must be understood and elucidated.

The second fundamental principle of Information Dynamics is that information has value. We claim that information only has value in *context*; in addition to context, value may also depend on usage, and may reflect other aspects as well. We need to better understand the role and properties of this value as we use information directly in our systems. Specifically, how does processing affect the value of information? How do movement, representation, and storage, affect its value? And what are the ramifications of this for system design?

The third fundamental principle of Information Dynamics is that the value of information changes with time. Typically it decreases but it may alternatively remain the same or even increase. Understanding the role time plays in the value of information has a clear impact on the applicability of information. Movement of information takes time. When the delay caused by movement becomes large, the impact on the value of information may be significant enough that further movement may be not only unnecessary, but may, in fact, be detrimental.

The principles of Information Dynamics presented here represent an attempt to capture a sufficient understanding of the fundamental characteristics of information to allow us to better design and implement complex systems. Although such an understanding has proven elusive, our efforts to date indicate that the proposed framework has the potential for bringing about a significant advancement in the way information is handled in systems.

REFERENCES

Bacsar, T. and Olsder, G. J., *Dynamic Noncooperative Game Theory*, Academic Press, New York, Mathematics in Science and Engineering, v.160, 1982.

Greenwald, A., “Modern Game Theory: Deduction vs. Induction,” TR 1998-756, New York University, 2/24/98.

Greenwald, A., “Learning to Play Network Games,” TR1998-758, New York University, 2/24/98.

Lamport, L., “Time, Clocks, and the Ordering of Events in a Distributed System,” *Communications of the ACM*, Vol. 21, No. 7, pp. 558-565, July 1978.

Owen, G., *Game Theory*, Saunders, Philadelphia, PA, 1968.

Stone, P. and Veloso, M., "Using Decision Tree Confidence Factors for Multi-Agent Control,” *Proceedings of the 2nd International Conference on Autonomous Agents*, ACM Press, NY, 1998, pp. 86-91.

Washington, R., "Markov Tracking for Agent Coordination,” *Proceedings of the 2nd International Conference on Autonomous Agents*, ACM Press, NY, 1998, pp. 70-77.