Cognition friendly interaction: A concept of partnership in human computer interaction

Balaram Das
Information Technology Division, DSTO, P.O. Box 1500, Salisbury, SA 5108, Australia

(Received 11 December 2000; accepted 8 May 2001; published 31 August 2001)

This paper identifies yet another field of research, the discipline of *human computer interaction*, where the concept of self-similar fluctuations can play a vital role. A concept of interaction between computation and cognition is developed that is friendly toward the cognitive process. It is argued that friendly interactions must have a memory and be antipersistent. To cast this in a mathematical form, fluctuations in the interactions recorded over a period of time are studied, and it is shown that these fluctuations must necessarily be self-similar with the value of the self-similarity parameter confined to the interval (0, 1/2), for the interaction to be friendly. A statistical measure of complexity, of the interaction process, is also formulated as a function of the self-similarity parameter. Finally the question is raised as how to build a friendly software and a possible evolutionary process through which friendly softwares may emerge is indicated. [DOI: 10.1063/1.1383548]

The French mathematician Henri Poincaré, while working on celestial mechanics about a hundred years ago, obtained an unique glimpse into the rich behavior of dynamical systems. This comprehension was not solely a product of his mathematical insight but the product of pure insight augmented by numerical calculations done laboriously by hand. The latter inconvenience has been removed, in recent times, by powerful computing softwares, enabling theoretical analysis and numerical investigation to proceed side by side and resulting in the spectacular advances in the understanding of dynamical systems that we see today. Is it possible to take a step further and forge a symbiosis between intuitive knowledge and computer generated understanding? A symbiosis that will enhance one's ability to explore and innovate, analyze and reflect, reason laterally, and even play a game of chess with inspiration. Recently some cognitive scientists have started pondering over these issues, issues which are no doubt difficult to come to grips with. For that reason, we start in this paper by asking—what would be the signature of such a symbiosis or when would we say that a problem solving process has been a cooperative process between cognition and computation? This question is tractable to some extent, as we show, by using the methods generally employed to investigate noise and fluctuation phenomena. Our work leads to further questions, the answers to which will provide valuable insight, and which we believe will yield to a probing with the tools of nonlinear analysis. In keeping with the theme of this focus issue we invite the readers of this journal to try out these questions.

I. INTRODUCTION

Self-similar fluctuations incorporating long range dependence have been associated with a wide range of phenomena; with finance and economics,[1] with the natural variability in physiological form and function,[2] and even with the traffic flow in the ethernet.[3] The cited bibliographical guide to self-similar processes[4] attests to the ubiquity of such fluctuations. Often self-similarity manifests itself as $1/f$ noise and the occurrence of such noise is well documented.[5] In this paper we exploit these concepts to develop a notion of friendly *human computer interaction* (HCI).

Research work in HCI has traditionally been dominated by the question of *user friendliness*. Here the focus is on understanding how people use the computer and therefore how to design better interfaces that make the computers user friendly. Good graphics, a smart use of color and visual display, speech recognition, etc., go a long way toward achieving this aim.[6] An important assumption made in these investigations is that the essential information processing is mainly done by the computer. The user feeds in the input and interacts with the computer in a routine manner until the final output is obtained. This does not take into account that, in problem solving processes, where computing softwares are used as aids, the user also processes a lot of information. Treating users as co-information-processors gives rise to the problems dealing with the cognitive interface in HCI, as opposed to the physical interfaces encountered in routine computing,[7] and highlights the need for computing packages to assist in the cognitive process. In other words softwares should not only be user friendly, but also cognition friendly, and it is this aspect of HCI we propose to investigate. Here by the terms *cognitive process* and *computing process* we mean the information processing carried out by a human and a computer, respectively.[7]

Let us state the problem along with its context. Suppose a *cognitive agent* (humans), in conjunction with a *computing agent* (computers), desires to undertake a certain problem-solving activity. This can be a real time interaction with some entity in the environment that is dynamic and complex and making decisions to guide its evolution. This can be an attempt to understand some natural phenomenon like turbulent flow, or it can just be playing a game of chess. In all of these, the cognitive agent's aim would be to resolve the problem at hand in a manner that is conducive to better comprehension or to further development along a desired direction. How would she interact with the computing agent in order to achieve this aim?

The essential point we are trying to identify and understand is a form of interaction between the process of cognition and the process of computation, which is equivalent to a partnership. The cognitive process is essentially different from the computing process. Computation is an algorithmic process that proceeds according to precise and consistent rules. Although modern computers are very efficient in processing long and complicated algorithms, this does not always help when one is engaged in performing complex tasks; sometimes these tasks cannot be performed algorithmically because no such algorithms exist, and sometimes the simplifications one introduces to obtain an algorithm are such that the task performed is significantly different from the intended one. Since the cognitive process is not hampered by the strict rules of computation, it is often more capable of handling such situations. One takes advantage of one's experiences, beliefs, intuitions, and even prejudices to arrive at inspired conclusions that are certainly not accessible through the process of computation.[8] We believe, when faced with complex tasks that require problem solving ability, the best strategy would be a cooperation between these different processes. A cooperation where computation is guided with cognitive insight and cognition is inspired by feedback from computed results. When such a partnership exists we say that the interaction is *cognition friendly*.

To develop a mathematical model of such interactions, we start by defining the concept of *dominance* in Sec. II. In Sec. III we study the fluctuations in accumulated dominance as observed in a series of interactions. A necessary condition, that such fluctuations must satisfy, for interactions to be friendly, is then deduced. In Sec. IV a measure of complexity is introduced which is statistical in nature and serves as a measure of correlation and partnership. In Sec. V we identify a process of evolution through which cognition friendly softwares may emerge. This leads to unanswered questions which are noted in the concluding Sec. VI.

II. DOMINANCE IN INTERACTION

The aim of the cognitive agent is to maneuver some aspect of the environment, which we call the *system (of interest)* and guide its evolution as desired. This is done in a manner similar to that in a multi-stage decision making process. To start with the cognitive agent determines the initial state of the system and exerts control to guide its evolution. The system is allowed to evolve for some time, after which its state is again ascertained and control exerted again. The state of the system at any time is represented as a point in the problem space or the state space; one starts from the initial state and guides the system toward a final goal state through a number of intermediate states called sub goal states.[9] Of course, the important part in this process is to decide upon the appropriate action at each stage. At any stage, the cognitive agent feeds in the data regarding the current state into the computing agent and obtains an output which indicates the course of action to be taken. The onus is now on the cognitive agent to decide whether or not to accept this course of action and therein lies the essence of the interaction.

At any stage only a part of the problem encountered might be comprehensible to the cognitive agent while the rest is incomprehensible. The latter part may consist of situations which require involved and long calculations or it may consist of situations about which the cognitive agent has no prior knowledge. Consider the comprehensible part first. The cognitive agent may or may not agree with the entire computed output for the comprehensible part or she may disagree with only a part of it, i.e., the computed course of action may not always appeal to the intuitive process. The cognitive agent would then accept only that part of the computed course of action with which she agrees. For the incomprehensible part the question of agreement with the computed course of action does not arise, as the cognitive agent has no intuitive understanding of the situation to start with. She may, however, decide to accept a part or the whole of the computed course of action. This will depend on the amount of faith the cognitive agent has in the abilities of the computing agent, and this in turn will depend upon the past performances of the computing agent.[10]

The accepted course of action at any stage will therefore consist of: (a) a part which is purely cognitive in origin; (b) a part which is suggested by the computing agent and with which the cognitive agent agrees; and (c) a part which is suggested by the computing agent but with which the cognitive agent is unable to agree or disagree. Let the fractions α, β, and γ correspond to parts (a), (b), and (c) of the total action: $\alpha+\beta+\gamma=1$. We now define the concept of dominance.

Dominance: At any stage of interaction, the dominance δ measures the amount by which the cognitive process has dominated over the computing process.

Clearly part (b) of the accepted course of action will not contribute to δ as both the agents agree over this part. Dominance will depend upon the amount by which part (a) exceeds part (c), hence we set $\delta=\alpha-\gamma$. δ can take all the values in the interval $[-1, 1]$, the negative values indicating the dominance of the computing process over the cognitive process. We stipulate that the cognitive agent exercise her judgement and assign values to α, β, and γ at every stage. This makes the process subjective but it is this subjectivity that is desired, for the concept of friendliness is a subjective concept. The value of dominance at each interaction should therefore reflect the subjective impression of the cognitive agent involved in that interaction.

Let the nth interaction take place at the point of time t_n, $0=t_1<t_2<\dots$; then we have a value of dominance defined at each point of time t_i. However, it may be argued that the time interval t_i-t_{i-1} may not be same for all values of i. This interval will in general be shorter at those points of time when the system changes faster. To avoid this, we stipulate that the cognitive agent decide upon an interval of duration η and control be exerted on the system when t takes the values

$0, \eta, 2\eta, \ldots$, etc. The duration η is chosen small enough so that the system is adequately controlled at all times. In fact t need not denote time at all; it can be any ordering that is naturally associated with the problem solving process in which case the question of unequal intervals may not even arise. The values of δ can now be considered to constitute a time series. δ takes the value δ^m, say, when t takes the value $m\eta$.

III. MODELING ACCUMULATED DOMINANCE

Let us now consider an experiment in which there are a number of pairs, each pair consisting of a cognitive agent working with a copy of the same computing agent engaged in controlling a copy of the same system. We assume that the cognitive agents involved have collectively decided upon a value for the constant η. The set of all the pairs constitute our sample space Ω and for each pair ω in this sample space the dominance of the cognitive agent over the computing agent is either positive or negative or zero at any point of interaction. If we consider a number of consecutive interactions, the *accumulated dominance* at the end of these interactions is given by adding the values of δ for each of these interactions. In meaningful interactions, the cognitive agent at each stage takes stock of what has happened up to that stage in order to decide upon a suitable action. Hence only the dominance as accumulated from the previous stages is essential. Let $X_\omega(t)$ be the variable denoting the accumulation of dominance, for the pair ω; then we have for $l \geqslant 1$, $X_\omega(t=l\eta)=\Sigma_{i=0}^{l-1}\delta_\omega^i$ and $X_\omega(0)=0$. Here δ_ω^i denotes the value of dominance for the pair ω at $t=i\eta$. If we now consider the set of all pairs and fix our attention on the interactions at $t=i\eta$, the collection of the values of dominance δ_ω^i, taken across all the pairs, at this point of time will constitute a random variable which we denote by $Y_{i\eta}$. More explicitly $Y_{i\eta}$ is a random variable defined on the sample space Ω which maps $\omega \rightarrow \delta_\omega^i$, $i=0,1,2,\ldots$.

For each sample ω we have the sample time series $X_\omega(t)$, and the ensemble of these series constitutes a stochastic process which we denote by $\boldsymbol{X}$. By $X(t)$ we denote another random variable defined on Ω which maps $\omega \rightarrow X_\omega(t)$ or, more explicitly,

$$X(t=l\eta)=\sum_{i=0}^{l-1} Y_{i\eta}, \quad l \geqslant 1, \quad \text{and} \quad X(0)=0. \tag{1}$$

The stochastic process $\boldsymbol{X}$ contains the statistical information regarding the accumulation of dominance; we propose to find a model for this process. We will build this model in three successive stages and to facilitate this process we recapitulate the following definitions.[11]

The stochastic process $\boldsymbol{X}$ is *self-similar* with the *self-similarity parameter H* if for any positive stretching factor s, the random variable $X(t)$ and the re-scaled random variable $s^{-H}X(st)$, have identical distribution.

The stochastic process $\boldsymbol{X}$ has *stationary increments* if for any increment, h, the random variable $X(t+h)-X(t)$ has distribution independent of t.

A. Modeled as Brownian motion

As a first step in modeling, consider the case where the interactions between the agents in a pair satisfy the following assumptions:

A1. The action taken by a pair at any stage is independent of the actions taken by any pair, including itself, on previous stages, i.e., the random variables $Y_0, Y_\eta, \ldots, Y_{l\eta}, \ldots$, are independent of each other.

A2. The random variables $Y_0, Y_\eta, \ldots, Y_{l\eta}, \ldots$, form a stationary sequence.

We will say more about assumption A1 later. Assumption A2, however, is plausible when one considers the fact that different pairs would adopt different ways to solve the same problem. At any point of time, therefore, different pairs would be facing different situations and there is no reason to believe that this variety would be differently constrained at different points of time. To make the formulation less cumbersome, we set, for all l, the mean $E[Y_{l\eta}]=0$ and the variance $E[(Y_{l\eta})^2]=\upsilon$, for some constant υ. From the central limit theorem it then follows that, at $t=k\eta$ for some large k, the accumulated dominance $X(t=k\eta)$ is a random variable which is normally distributed with a mean *zero* and variance proportional to t, we write the variance as $\sigma^2 t$; $\sigma^2=\upsilon/\eta$. Similarly for $h=k\eta$ and large k, the increment $X(t+h)-X(t)$, is normally distributed with mean *zero* and variance $\sigma^2 h$. Furthermore, the increments $X(\eta)-X(0)$, $X(3\eta)-X(2\eta), \ldots$, are random variables which are independent of each other. To model this process we seek a continuous time stochastic process that will have all of these characteristics for all finite values of t. Up to now we have been measuring dominance at discrete time points $0, \eta, 2\eta, \ldots$, etc. As we pass on to the limit $\eta \rightarrow 0$, $X(t)$ in Eq. (1) has to be constructed as a normalized sum of the random variables $Y_{l\eta}$, in order for it to remain finite for finite t. Such a construction already exists in the form of the *Wiener process* or the *Brownian motion*.[11,12] Brownian motion is the limiting distribution of the sum of normalized step lengths in a random walk. Hence as our first model of the accumulation of dominance we replace the discrete process by the continuous time Brownian motion $\boldsymbol{X}$, defined on the sample space Ω. The corresponding random variable $X(t)$ now satisfies the following:

(BM1) With probability1, $X(0)=0$, value of accumulated dominance is zero to start with.

(BM2) For any $t \geqslant 0$ and $h>0$ the increment $X(t+h)-X(t)$ is normally distributed with mean 0 and variance $\sigma^2 h$ or

$$P((X(t+h)-X(t)) \leqslant x) = \frac{1}{\sigma\sqrt{2\pi h}} \int_{-\infty}^{x} \exp(-u^2/2\sigma^2 h)\, du. \tag{2}$$

(BM3) If $0 \leqslant t_1 \leqslant t_2 \ldots \leqslant t_{2m}$, then the increments $X(t_2)-X(t_1), X(t_4)-X(t_3), \ldots, X(t_{2m})-X(t_{2m-1})$ are independent.

Here $P(A)$ as usual denotes the probability of the event A and σ^2 is the constant defined before.

In a real problem solving process, however, the dominance at any stage will depend upon the nature of interaction that precedes it. Moreover, a kind of interaction that is independent of the past will certainly not represent a friendly interaction, as can be seen by analyzing interactions between humans. Friendly interactions evolve over a period of time, future interactions depend upon the past, or the process incorporates a memory. Furthermore, if a problem is cognitively transparent then there is no need to interact with a computing agent friendly or otherwise except for routine computations; similarly, it would be a waste of cognitive effort not to compute the solution of a problem straight away if a competent algorithm exists. In dealing with problems that do not admit such ready resolution, the cognitive agent needs to interact with the computing agent to clarify the problem at hand. In friendly interactions an amount of computation leads to an amount of cognitive transparency. This will in turn induce cognitive action. The cognitive agent will be encouraged to try novel methods which are more efficient or are better able to tackle the inherent difficulties of the problem. For the same reason, after an amount of cognitive effort the cognitive agent will feel a need for computation in order to gain further insight. In other words, in a friendly interaction neither cognition nor computation should dominate persistently. This again is true for interactions between humans; long term cooperative interactions exist where the participants need help from each other and none dominate persistently. More to the point, if we want to ascertain that a person is friendly, then we study the interactions of this person with a number of different persons over a period of time, and determine if the statistical nature of all these interactions incorporates memory and does not show a persistent accumulation of dominance. Therefore, to model the accumulation of dominance in a cognition friendly interaction, we necessarily need a variation of the Brownian motion that has (i) a *memory* and is (ii) *antipersistent*: the meaning of these two terms will be made precise in the following section.

B. Modeled as a self-similar process with stationary increments

It is easily shown that Brownian motion is a self-similar process with self-similarity parameter $H=1/2$, and that it has finite variance and stationary increments. Furthermore as stipulated in (BM3) the increments are independent. In fact Brownian motion is the unique Gaussian process that has these properties.[12] For our purpose, the simplest variation would be one which retains all the above characteristics but drops the independence of increments, i.e., drops assumption (A1). Can such a process be constructed which can serve as a model of the accumulation of dominance? The answer is in the affirmative according to a limit theorem by Lamperti[11,13] and we have the following:

> Let us drop assumption A1 but retain A2, i.e., Y_0, $Y_\eta,\ldots,Y_{l\eta},\ldots$, form a stationary sequence of random variables but are not necessarily independent of each other. Lamperti's theorem now affirms that there exists a continuous time stochastic process $\boldsymbol{X}$, which models accumulated dominance, with the property that this process $\boldsymbol{X}$ is self-similar and has stationary increments, and that the self-similarity parameter $H>0$.

To explore the properties of this self-similar process $\boldsymbol{X}$ we set $X(0)=0$ with probability 1, implying that value of the dominance is zero to start with, and since the process is self-similar we can set $E[X(t)]=0$ for all t. We also set $\sigma^2=E[(X(t+1)-X(t))^2]=E[(X(1))^2]$; the second equality follows from the stationarity in increments. The following are now easily established.[11] The variance of a general increment at any time t and for any $h>0$, is given by

$$E[(X(t+h)-X(t))^2]=E[(X(t)-X(t-h))^2]=\sigma^2h^{2H}. \tag{3}$$

Similarly, at any time t and for any $h_1,h_2>0$ covariance between a past increment $X(t)-X(t-h_1)$ and a future increment $X(t+h_2)-X(t)$ is given by

$$E[(X(t)-X(t-h_1))(X(t+h_2)-X(t))] = \frac{\sigma^2}{2}((h_1+h_2)^{2H}-h_1^{2H}-h_2^{2H}). \tag{4}$$

To put a bound on the value of the self-similarity parameter, consider the sequence of increments in dominance: $D_i=X(i)-X(i-1),\ i\geqslant 1$. The covariance between D_i and $D_{i+k},\ k>0$, is given by

$$\chi(k)=E[D_iD_{i+k}]=E[D_1D_{1+k}] = \frac{\sigma^2}{2}[(k+1)^{2H}-2k^{2H}+(k-1)^{2H}]. \tag{5}$$

When $H=1$, $\chi(k)=\sigma^2$, and when $H>1$, $\chi(k)$ increases monotonically with k. Hence when $H\geqslant 1$, the correlation between the increments in accumulated dominance calculated at two different points of time either remains constant or grows indefinitely as the two points move farther apart. This can hardly be the case with meaningful interactions. We therefore confine our attention to values of H in the interval $0<H<1$. For $H=1/2$, $\chi(k)=0$, indicating that future and past are uncorrelated. For all other values of H in the interval (0, 1), $\chi(k)$ is positive for finite values of k and tends to *zero* as $k\rightarrow\infty$. More precisely, $\chi(k)/(H(2H-1)k^{2H-2})\rightarrow 1$ as $k\rightarrow\infty$. Hence the self-similar process $\boldsymbol{X}$ can only represent the accumulation of dominance in meaningful interactions if the self-similarity parameter H lies in the interval (0, 1).

At any point t the nature of correlation between the past and future increments are given by the value of covariance in Eq. (4). This value is zero only when $H=1/2$ (Brownian motion); for all other possible values of H it is nonzero, indicating that future increments are correlated to the past increments. One generally says that the process has a *memory* when $H\neq 1/2$; our first requirement for friendly interaction. When $0<H<1/2$, the covariance is negative and this is referred to by saying that the process is *antipersistent*. In terms of dominance this means that, on the average, if at any point of time t it is found that one of the participants in a pair has dominated in the recent past, then there will be a tendency for the other participant to dominate in the near future. This satisfies our second requirement. Hence we conclude the following:

> *A statistical model of the accumulation of dominance in a cognition friendly interaction is a continuous time stochastic process* $\boldsymbol{X}$ *which is self-similar with the self-*

similarity parameter H confined to 0<H<1/2, and which has stationary increment.

When $1/2<H<1$ the covariance in Eq. (4) is positive and we say that the process incorporates *persistent* dominance.

The fact still remains that $\chi(k)$ is nonzero for arbitrarily large values of k. Does this mean that at any point of time a cognitive agent takes into account all past interactions in order to conduct the present interaction or is this fact an aberration arising from our modeling technique? The question is answered by considering the central property of our model: that the process is self-similar. To see this consider restating self-similarity as follows. $\boldsymbol{X}$ is self-similar is equivalent to the statement

$$P(X(t+h)-X(t)\leqslant x)=P(s^{-H}X(st+sh)-X(st)\leqslant x),\quad s>0.$$

In other words the statistical nature of the stochastic process $\boldsymbol{X}$ when considered over a time interval $(t,t+h)$ is indistinguishable from the statistical nature of the stochastic process $s^{-H}\boldsymbol{X}$ when considered over the time interval $(st,\ st+sh)$. Interactions over a larger interval when viewed in lesser detail would have the same characteristics as interactions over a smaller interval when viewed in greater detail. Hence a cognitive agent does not have to remember the past in all its exact details in order to influence the present and the near future. Past interactions over longer intervals of time need only be captured with lesser details. Furthermore in contrast to deterministic self-similarity where the same pattern repeats itself *exactly* at various scales, in a statistically self-similar process what remains same at various time scales is not the exact pattern of the process but the general trend of the process. Again what needs to be remembered by a cognitive agent is the general trend or the qualitative nature of the pattern of past interactions. Depending upon their abilities, different cognitive agents will capture this pattern to different levels of approximations. These aspects, we may note, are also the predominant aspects in the interactions of humans with fellow humans.

The other central characteristic of our model stipulates that the process has stationary increments. In other words, observation of the accumulation of dominance during some time interval of length h will only reveal the statistical nature of dominance in this interval; it will not reveal the point of time when the observations were made, or the collective behavior of the pairs, considered at different points of time, are indistinguishable. In other words, as time progresses the intrinsic character of the problem-solving process remains the same. This again is quite natural because of the following reasons. First, recall that the nature of the computing agent does not change while solving a given problem. Second, it is unlikely that, on average, the cognitive capabilities of the cognitive agents would significantly change during a single problem-solving process. Hence the central characteristics of our model that of self-similarity and of stationary increments only reflect some of the fundamental aspects of interactions which involve humans.

C. Modeled as fractional Brownian motion

If the random variables $Y_0, Y_\eta, \ldots, Y_{l\eta}, \ldots$, are weakly dependent, then it is possible for $\boldsymbol{X}$ to be Gaussian, whereas strong dependency can sometimes result in a non-Gaussian process.[14] Although this is not an impediment for most of the rest of this article, some of the points we make will not hold strongly for non-Gaussian processes. We therefore make the following modification to our model. Let us require that in addition to possessing the properties described in Sec. B, the stochastic process $\boldsymbol{X}$ modeling the accumulated dominance also be Gaussian. It then follows that $\boldsymbol{X}$ is a *fractional Brownian motion* (*FBM*) as proposed by Mandelbrot and Van Ness.[15] In fact *FBM* is the unique Gaussian process that has finite variance and stationary increments, and is self-similar with the self similarity parameter $H\in(0,1)$.[11] More explicitly, if the process $\boldsymbol{X}$ is an *FBM*, then the corresponding random variable $X(t)$ satisfies the following:[12]

(FBM1) with probability 1, $X(0)=0$,

(FBM2) for any $t\geqslant 0$ and $h>0$ the increment $X(t+h)-X(t)$ is normally distributed with mean 0 and variance $\sigma^2 h^{2H}$ or

$$P(X(t+h)-X(t)\leqslant x)=\frac{1}{\sqrt{2\pi}\sigma h^H}\int_{-\infty}^{x}\exp(-u^2/2\sigma^2h^{2H})\,du.$$

Hence requiring that $\boldsymbol{X}$ be Gaussian leads to:

The statistical model of the accumulation of dominance in a cognition friendly interaction, is an FBM with the self-similarity parameter H confined to the interval 0<H<1/2.

The following question now arises. Suppose we conduct the experiment as discussed above, i.e., we take a number of pairs, each consisting of a cognitive agent in conjunction with a copy of the same computing agent and obtain the time series for accumulated dominance for each pair. How would we know that the interactions recorded are friendly interactions? Let $S_X(f)$ be the power spectrum of $\boldsymbol{X}$: it is generally defined as the Fourier transform of the 2-point autocorrelation function $E[X(t+h)X(t)]$. For nonstationary processes this is not well defined. One can, however, use filtering-squaring-averaging operations to obtain an estimate of $S_X(f)$. If $\boldsymbol{X}$ is an *FBM*, then for all practical purposes it is taken for granted that $S_X(f)\propto 1/f^u$, where u and H are related as $u=2H+1$.[12,15] Hence if the stochastic process that results from our experiment reflects a cognition friendly interaction, then the spectral density function will obey the power law

$$S_X(f)\propto 1/f^u \quad \text{with}\quad 1<u<2. \tag{6}$$

IV. STATISTICAL COMPLEXITY OF THE INTERACTIONS

We now seek to understand the influence of the self-similarity parameter on the intrinsic nature of the interaction. Here by intrinsic nature we mean *complexity* which encompasses factors like dominance, friendliness, memory etc., i.e., factors that are fundamental to this analysis. We have up to now portrayed the interaction as persistent or antipersistent only. Can we take the next step and quantify the complexity of the interaction process and see how this varies with H? In the recent past a great many measures of complexity have been proposed. These reflect the attempts by various workers to quantify their intuitive understanding of structures, patterns, and relationships ingrained in interesting systems and processes. Of course, the notions of structure and pattern are intimately related to the more basic notions of order and disorder. Measures of complexity importantly, therefore, capture a measure of order or disorder. Some complexity measures increase with order, while others increase with disorder.[16] However, the measures that are intuitively most satisfying are those that assign very low values of complexity to systems that are either completely ordered or completely disordered. This point of view holds that a completely ordered system like a perfect crystal and a completely disordered system like a box of gas molecules in equilibrium are systems which have virtually no complexities.[17] Complex systems are positioned in between order and chaos and this somehow imparts them interesting characteristics like the ability to adapt to the environment and be robust and fault tolerant. An important class of complexity measures that vanish in the extreme order and disorder limit are termed as *statistical complexity* measures. The word statistical serves the obvious purpose of distinguishing these measures from deterministic complexities such as the Kolmogorov–Chaitin complexity.[18] A number of statistical complexity measures have been proposed in the literature, again these strive to measure intrinsic features like patterns, correlations, self-organization, etc. We refer the reader to Ref. 19 for a review and discussion of such measures. The measure of complexity we propose is statistical in nature and serves as a measure of correlation and partnership: correlation between past and future and partnership between cognition and computation.

Suppose we have made observations over the duration $[0,T]$. Then a measure of correlation between the past and the future can be obtained by fixing our attention at the point of time $T/2$ and, relative to this point, evaluating the correlation between the accumulation of dominance in the past which is $\{X(T/2)-X(0)\}$, and the future accumulation of dominance, which is $\{X(T)-X(T/2)\}$. A generalization of this concept would be to fix our attention at any point of time t and measure the correlation between the random variables $\{X(t)-X(t-h)\}$ and $\{X(t+h)-X(t)\}$, for some duration h. Such a correlation is given by

$$\rho=\frac{E[(X(t)-X(t-h))(X(t+h)-X(t))]}{\{E[(X(t)-X(t-h))^2]E[(X(t+h)-X(t))^2]\}^{1/2}}$$
$$=2^{2H-1}-1. \tag{7}$$

The last step follows from Eqs. (3) and (4). It is interesting to note that ρ is independent of both t and h and depends upon the self-similarity parameter H only, hence we feel justified in using it as the measure of the *intrinsic correlations* ingrained in the interactions. $H=1$ implies $\rho=1$, corresponding to the situation of perfect correlation. $H=1/2$, on the other hand, implies $\rho=0$, future and past are uncorrelated; in case of an *FBM* these are independent as the process reduces to Brownian motion.

As for partnership, consider a pair ω in the sample space Ω and the corresponding time series $X_\omega(t)$, which records the accumulation of dominance for this pair. If $X_\omega(t)$ is positive at some point of time t, then on balance, the cognitive agent has had a dominant contribution to the problem-solving process in the interval $[0,t]$, and a similar conclusion holds for the computing agent when $X_\omega(t)$ is negative. Hence a measure of partnership between cognition and computation can be obtained by measuring the frequency with which $X_\omega(t)$ changes sign. The *graph* of $X_\omega(t)$ is nothing but the set of points $\{t,X_\omega(t)\}$ in R^2 and the *zeroset* of $X_\omega(t)$ is the set of points at which the graph of $X_\omega(t)$ intersects the time axis. Since $X_\omega(t)$ is a sample function of a self-similar stochastic process, it follows that its zeroset is self-similar and hence has a well defined Hausdorff and box dimension.[12] It can be shown under very general conditions that, with probability 1, the sample graph of a self-similar process with self-similarity parameter H has a Hausdorff dimension[12,20] $(2-H)$. Hence the zeroset of $X_\omega(t)$ for any pair ω has a Hausdorff dimension

$$d=1-H. \tag{8}$$

We take this Hausdorff measure d as a measure of *partnership* between cognition and computation in any pair. Note that partnership is a measure and is defined for each pair in the ensemble of all pairs. Since it is the same for each pair, it can also be thought of as a measure pertaining to the entire ensemble of pairs. On the other hand, persistence and antipersistence, as discussed in Sec. III B, are not quantitative measures, they qualify the nature of interaction and are only defined for the ensemble taken as a whole. However, one can think of a degree of persistence and a degree of antipersistence in terms of partnership, or the values of d, in the following manner. We say that the degree of antipersistence increases as H decreases from 1/2 to 0, because in this case d increases from 1/2 to 1. Similarly the degree of persistence increases as H increases from 1/2 to 1, because in this case d decreases from 1/2 to 0. Hence $H=1$ and $H=0$ represent, respectively, the highest levels of persistence and antipersistence within our range of inquiry.

Our measure of complexity depends upon the measures of intrinsic correlation and partnership and we formulate it as:

$$C=|\rho|d=|2^{2H-1}-1|(1-H). \tag{9}$$

Figure 1 shows the behavior of C against the self-similarity parameter. $C=0$ when $H=1$. This we recall is the situation of perfect correlation; here the random variables representing past and future increments are linearly related or

$$|\rho|=1\Leftrightarrow\{X(t+h)-X(t)\}=a\{X(t)-X(t-h)\}+b,$$

for some constants a and b. At any point of time, therefore, the future dominance is related to the past dominance by the same linear law for every pair. This rigidity in pattern is brought about by the fact that $H=1$ also corresponds to the highest level of dominance. There are no interactions in this case. One agent in each pair dictates the problem-solving process. As a result the complexity C which quantifies partnership in interactions vanishes. C also vanishes when H

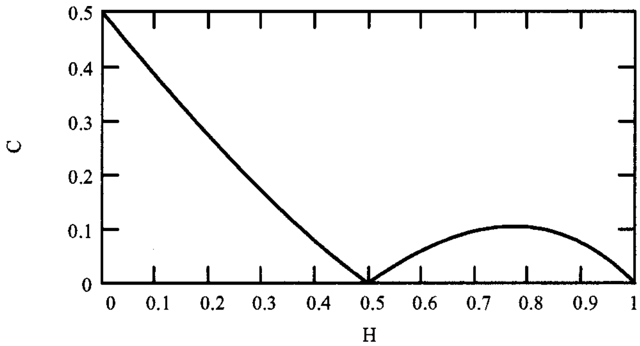

FIG. 1. Measure of complexity C as a function of the self-similarity parameter H. C vanishes when past and future dominance are perfectly correlated, i.e., $H=1$, and when they are independent, i.e., $H=1/2$, and is nonzero elsewhere. For friendly interactions, C which measures correlation and partnership increases as H decreases from 1/2 to 0. $H=1$ and $H=0$ represent, respectively, the highest levels of persistence and antipersistence within our range of inquiry.

$=1/2$ and this is due to the fact that future and past interactions are now independent. Although the agents interact, the interaction incorporates no structure. Finally for friendly interactions the complexity increases as H decreases from 1/2 to 0.

V. EVOLUTION OF A FRIENDLY SOFTWARE: A RESEARCH PROPOSAL

In keeping with the theme of this special issue, we now develop a line of inquiry that will lead to questions for the general reader. We have formulated the concept of cognition friendliness and have discussed its characteristics, the question that naturally arises is: how can a software be built that will exhibit friendliness? The attribute of cognition friendliness is most likely to be achieved through a process of evolution. In fact, many systems that exhibit *1/f* fluctuations in some of their parameters are systems that have evolved with time.[5] Hence a more pertinent question would be: *what manner of evolution must a software undergo in order to achieve cognition friendliness*? In the following we sketch a possible path toward an answer, or offer a research proposal to tackle the above question, by indicating how evolutionary processes may be associated with the process of friendly interaction.

Our proposal exploits the concept of *self-organized criticality* as formulated by Bak *et al.*[21] The canonical example they give considers building a pile of sand by adding one grain at a time at random positions. As the pile grows the addition of new grains causes big and small avalanches. At some point the pile ceases to grow: additional grains only cause other grains to fall off the pile — the pile has reached a statistically stationary state. At this point addition of new grains will cause avalanches of all sizes, possible for the system, with power law spatial and temporal distribution functions. The pile is said to have achieved a self-organized critical state. Analogous to this process, let us consider the following manner of evolution of a software, in which we expect it to achieve cognition friendliness when it has evolved to a stationary state. Choose a particular problem and a computing software appropriate for this problem. This software need not have the capability to tackle all aspects of the problem chosen, nor need it be a friendly software. All we want is an initial computing agent to start with. Let a number of experts work with this initial software, each interacting with a copy of his or her own. The experts work with the software and add new modules in order to add new functionality and enhance the already existing ones. Every time an expert adds a new module, she does so that the computation carried out by this module clarifies some aspect of the problem which she has encountered. Moreover, when an expert decides upon a new module, all copies of the software are simultaneously updated by the inclusion of this new module. Different experts will add different modules because they have different ways of solving the same problem. The software will undergo a process of evolution. What it evolves into will depend upon what modules are added and most importantly how they are linked to each other.

Now, the basic software can only be improved as much as the experts are capable of. For example, mathematicians at present have a certain degree of understanding of turbulent flow. Any software being created to help solve problems in turbulent flow can only be as good as this current understanding. More generally, at any stage of problem solving, the cognitive agents collectively will have a certain degree of understanding of the problem. The limit to this understanding will put a limit on the evolution of the software. In other words, a stage will come when addition of further modules will not add new functionality, because all known aspects of the problem have already been covered. When this happens we say that the software has evolved to a *stationary state*. As the software evolves the process of interaction with the software also evolves. *Will this interaction evolve to a friendly one when the software evolves to a stationary state?* At this point one may wonder as to why we think that stationary softwares would lead to friendly interaction. To clarify this let us analyze the interaction with the stationary software.

Consider the state of affairs where the computing agent in each pair is the stationary software. Suppose that the problem-solving activity has been divided in to a number of sub tasks. At any stage the action taken will be with regard to one or more sub tasks that are being performed at that stage. Suppose the kth sub task T^k is initiated at time t_1 by the pair ω. Let $X_\omega^k(t)$ be the contribution to the accumulated dominance $X_\omega(t)$ by this subtask. As time increases the accumulation $X_\omega^k(t)$ will increase whenever the cognitive agent dominates and decrease whenever the computing agent dominates with regard to the sub task T^k. For those stages where no action is taken with regard to T^k, or where there is agreement between the agents, this accumulation will remain unchanged. At time t_1 the accumulation $X_\omega^k(t)$ is zero because the task T^k is initiated at this point. Suppose this accumulation increases and then decreases to become zero again at time t_2. In analogy with the sand pile model, we say that there has been a k-type avalanche of positive accumulated dominance $X_\omega^k(t)$ having a lifetime (t_2-t_1). Clearly, by the time t_2, the computing agent has started to dominate as far as the sub task T^k is concerned. From time t_2 onward the accumulation $X_\omega^k(t)$ will most probably proceed through negative values and this negative accumulation will rise up toward zero when the cognitive agent starts dominating again. Suppose the zero value is attained again at the time t_3; we then have a *k*-type avalanche of negative accumulated dominance having a lifetime (t_3-t_2). This is a computation dominated avalanche while the former was cognition dominated.

The total accumulated dominance $X_\omega(t)$ at any point t for any pair is nothing but the net accumulation obtained by a linear superimposition of all types of avalanches, corresponding to different sub tasks that are active at the point *t*. These avalanches will have all possible lifetimes. Christensen *et al.*[22] have studied the avalanches that are created when the stationary sand pile is perturbed by adding grains of sand. To each avalanche they associate a dissipation rate. The total dissipation rate $j(t)$ at any given time t is given by the linear superposition of the individual dissipation rates produced by the individual avalanches operating at time *t*. In fact $j(t)$ in their context is similar to $X_\omega(t)$ in our context. They show that for the stationary sand piles the power spectrum of total dissipation obeys a power law: this is the signature of self-organized criticality. We therefore expect the stochastic process $\mathbf{X}$ to have a power spectrum $S_X(f) \propto 1/f^u$ when the software used by each pair is the stationary software. When this happens we say that the interaction has evolved to a state of self-organized criticality. If in addition u lies in the interval $1<u<2$, then according to Eq. (6) the interaction is cognition friendly.

VI. CONCLUSIONS AND OPEN QUESTIONS

The evolution of the software will depend upon (a) the modules added by various experts and (b) the manner in which the modules are interlinked. We now summarize the line of inquiry outlined in Sec. V and pose the following question to the interested reader.

Will the manner of evolution proposed above lead to a friendly software? More precisely, as the software evolves to a stationary state, will the interaction with the software evolve to a state of self-organized criticality with the correct power spectrum, i.e., will $\mathbf{X}$ evolve to have $S_X(f) \propto 1/f^u$ with $1<u<2$?

A definite answer can only be obtained by selecting a problem and building the corresponding software in the manner proposed. In this regard it is interesting to take note of the conclusions arrived at by Bak and others. From their many experiments they conclude that the critical state with *1/f* behavior is unavoidable,[23] and does not require fine tuning of external parameters, or that such a state is an attractor. Considering the method we have used for the evolution of the software, an attractor is more likely to be a state that is a cognition friendly state than one that is not.

We also note that experts gain experience by interacting with a stationary software. As this experience grows, the intuitive understanding of the problem may also grow and aspects of the problem yet unknown will come to light. This will enable addition of new modules not possible before and the software will transit to a new stationary state in keeping with the better intuitive understanding, and so on.

A software engineer will need to sort out technical details in addition to the concepts discussed above to build a friendly software. One has to figure out not only how to link the modules to build the software, but also how to prune the software to remove modules which may have ceased to be of relevance.[24] Like many systems that have undergone evolution, the friendly software will necessarily have as a feature an amount of redundancy. Different modules may be functionally very similar but are there because different experts prefer different ways to tackle the same problem. This is, however, different from the case where for some reason a module has lost relevance and is not being used, perhaps because problem-solving methods have been improved in general. To prevent the software from becoming unnecessarily unwieldy a marker can be set up which monitors the modules and expunges those which have not been used for a preset length of use.

Finally, this paper only takes the first step in introducing the concept of cognition friendliness and defining it in terms of quantities that can be measured in an experiment. We believe the model developed can form a basis to start formulating further meaningful questions regarding the notion of friendliness between humans and computers and devising methods for building friendly computer softwares.

ACKNOWLEDGMENT

The author wishes to thank John Hansen for many illuminating discussions and clarifications.

[1] B. B. Mandelbrot, *Fractals and Scaling in Finance*, 1st ed. (Springer-Verlag, New York, 1997).
[2] B. J. West, Ann. Biomed. Eng. **18**, 135–149 (1990).
[3] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, IEEE/ACM Transactions on Networking **2**, 1–15 (1994).
[4] M. S. Taqqu, in *Dependence in Probability and Statistics,* edited by E. Eberlein and M. S. Taqqu (Birkhauser, Boston, MA, 1986), pp. 137–165.
[5] M. S. Keshner, Proc. IEEE **70**, 212–218 (1982); M. Kobayashi and T. Musha, IEEE Trans. Biomed. Eng. **BME-29**, 456–457 (1982); M. B. Weissman, Rev. Mod. Phys. **60**, 537 (1988); P. Dutta and P. M. Horn, *ibid.* **53**, 497 (1981).
[6] B. Shneiderman, in *Human-Computer Interaction: Toward the Year 2000,* 2nd ed., edited by R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg (Morgan Kaufmann, San Francisco, CA, 1995), p. 399.
[7] S. K. Card and T. P. Moran, in *Human-Computer Interaction: Toward the Year 2000*, 2nd ed., edited by R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg (Morgan Kaufmann, San Francisco, CA, 1995), Chap. 9, p. 587; see also the preface to chapter 9 for a review and articles by S. Card, T. Moran, and A. Newell, cited therein.
[8] R. Penrose, *The Emperor's New Mind,* 1st ed. (Oxford University Press, Oxford, 1989), Chap. 10.
[9] A. Newell and H. A. Simon, *Human Problem Solving* (Prentice-Hall, Englewood Cliffs, NJ, 1972).
[10] As queried by an anonymous referee, consider the following question as an example: what happens if at a certain stage the problem encountered involves inconsistencies? Computation being an algorithmic process cannot handle inconsistencies. The problem would have to be simplified, say by adopting the methods of fuzzy logic or other such methods of approximation. Thus reformulated, the problem can be submitted to the computing agent and a solution obtained. A cognitive agent, on the other hand, can have some understanding of a problem even if it involves inconsistencies. The part which she intuitively understands will come under the comprehensible part, while the part about which she has no intuitive understanding will come under the incomprehensible part. For the comprehensible part the intuitive solution may not completely agree with the computed solution: the cognitive agent would then accept only that part of computed course of action with which she agrees. For the incomprehen-

sible part, the cognitive agent can accept a part or the whole of the computed course of action, depending on how confident she feels about the approximations performed.

[11] J. Beran, *Statistics for Long-Memory Process* (Chapman & Hall, New York, 1994), Chap. 2.

[12] K. Falconer, *Fractal Geometry* (Wiley, Chichester, 1990), Chap. 16.

[13] J. Lamperti, Trans. Am. Math. Soc. **104**, 62–78 (1962).

[14] M. S. Taqqu, in *Encyclopedia of Statistical Sciences*, edited by S. Kotz and N. Johnson (Wiley, New York, 1988), Vol. 8, pp. 352–357.

[15] B. B. Mandelbrot and J. W. Van Ness, SIAM Rev. **10**, 422–437 (1968).

[16] J. S. Shiner, M. Davison, and P. T. Landsberg, Phys. Rev. E **59**, 1459–1464 (1999).

[17] S. Lloyd and H. Pagels, Annals of Physics **188**, 186–213 (1988); B. A. Huberman and T. Hogg, Physica D **22**, 376–384 (1986).

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley, New York, 1991), Chap. 7.

[19] D. P. Feldman and J. P. Crutchfield, Phys. Lett. A **238**, 244–252 (1998).

[20] N. Kono, in *Dependence in Probability and Statistics*, edited by E. Eberlein and M. S. Taqqu (Birkhauser, Boston, MA, 1986), pp. 109–117.

[21] P. Bak, C. Tang, and K. Wiesenfeld, Phys. Rev. Lett. **59**, 381–384 (1987).

[22] K. Christensen, H. C. Fogedby, and H. J. Jensen, J. Stat. Phys. **63**, 653–684 (1991).

[23] P. Bak, K. Chen, and M. Creutz, Nature (London) **342**, 780–782 (1989); P. Bak, *How Nature Works* (Copernicus, Springer-Verlag, New York, 1996).

[24] This aspect was pointed out by an anonymous referee.