A Complex Systems Perspective on Computer-Supported Collaborative Design Technology

Mark Klein
Massachusetts Institute of Technology
m_klein@mit.edu

Hiroki Sayama
New England Complex Systems Institute
sayama@necsi.org

Peyman Faratin
Massachusetts Institute of Technology
peyman@mit.edu

Yaneer Bar-Yam
New England Complex Systems Institute
yaneer@necsi.org

The Challenge: Collaborative Design Dynamics

Collaborative design is challenging because strong interdependencies between design issues make it difficult to converge on a single design that satisfies these dependencies and is acceptable to all participants. Current collaborative design processes are typically characterized by (1) multiple iterations and/or heavy reliance on multi-functional design reviews, both of which are expensive and time-consuming, (2) poor incorporation of some important design concerns, typically later life-cycle issues such as environmental impact, as well as (3) reduced creativity due to the tendency to incrementally modify known successful designs rather than explore radically different and potentially superior ones.

This article examines what complex systems research can do to help address these issues, by informing the design of better computer-supported collaborative design technology. We will begin by defining a simple model of collaborative design, discuss some of the insights a complex systems perspective has to offer, and suggest ways to better support innovative collaborative design building on these insights.

Defining Collaborative Design

A design (of physical artifacts such as cars and planes as well as behavioral ones such as plans, schedules, production processes or software) can be represented as a set of *issues* (sometimes also known as *parameters*) each with a unique value. If we imagine that the possible values for every issue are each laid along their own orthogonal axis, then the resulting multi-dimensional space can be called the *design space*, wherein every point represents a distinct (though not necessarily good or even physically possible) design. The choices for each design issue are typically highly *interdependent*. Typical sources of inter-dependency include shared resource (e.g. weight, cost) limits, geometric fit, spatial separation requirements, I/O interface conventions, timing constraints etc.

Collaborative design is performed by multiple participants (representing individuals, teams or even entire organizations), each potentially capable of proposing values for design issues and/or evaluating these choices from their own particular perspective (e.g. manufacturability).

Some designs are better than others. We can in principle assign a *utility* value to each design and thereby define a *utility function* that represents the utility for every point in the design space. The *goal* of the design process can thus be viewed as trying to find the design with (close to) the optimal (maximal) utility value,.

The key challenge raised by the collaborative design of complex artifacts is that the design spaces are typically huge, and concurrent search by the many participants through the different design subspaces can be expensive and time-consuming because design issue interdependencies lead to conflicts (when the design solutions for different subspaces are not consistent with each other). Such conflicts severely impact design utility and lead to the need for expensive and time-consuming design rework.

Insights from Complex Systems Research

A central focus of complex systems research is the dynamics of distributed networks, i.e. networks in which there is no centralized controller, so global behavior emerges solely as a result of concurrent local actions. Such networks are typically modeled as multiple nodes, each node representing a state variable with a given value. Each node in a network tries to select the value that optimizes its own utility while maximizing its consistency with the influences from the other nodes. The global utility of the network state is simply the sum of node utilities plus the degree to which all the influences are satisfied. The dynamics of such networks emerge as follows: since all nodes update their local state concurrently based on their current context (at time T), the choices they make may no longer be the best ones in the new context of node states (at time T+1), leading to the need for further changes.

Is this a useful model for understanding the dynamics of collaborative design? We believe that it is. It is straightforward to map the model of collaborative design presented above onto a network. We can map design participants onto nodes, where each participant is trying to maximize the utility of the choices it is responsible for, while ensuring its decisions will satisfy the relevant dependencies (represented as the links between nodes). As a first approximation, it is reasonable to model the utility of a design as the local utility achieved by each participant plus a measure of how well all the decisions fit together. Even though real-world collaborative design clearly has top-down elements, the sheer complexity of many design artifacts means that no one person is capable of keeping the whole design in his/her head and centralized control of the design decisions becomes impractical, so the design process is dominated by concurrent local activities. The remainder of this paper will be based on this view of the collaborative design process.

How do such distributed networks behave? Let us consider the following simple example, a network consisting of inter-linked binary-valued nodes. At each time step, each node selects the value for itself that is the same as that of the [majority of the] nodes it is linked to. We can imagine using this network to model a real-world situation wherein six subsystems are being designed and we want them to use matching interfaces. The network has converged onto a *local* optimum (no node can increase the number of influences it satisfies by a local change), so it will not reach as a result a *global* optimum (where all the nodes have the same value). (Figure 1):

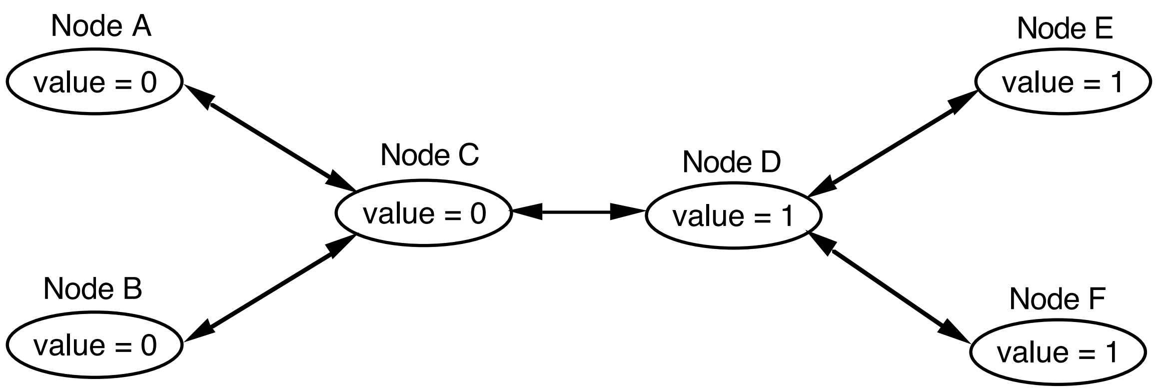

Figure 1: A simple network.

Generally speaking, networks may not always converge upon the global optimum, and in some cases (as we shall see with *dynamic attractors*), a network may not converge at all. *Insights into whether and how global optima can be found in networks represent the heart of what complex systems research offers to the understanding of collaborative design.*

We will discuss these insights in the remainder of this section. The key factor determining network dynamics is the nature of the influences between nodes. We will consider two important distinctions: whether the influences are *linear* or not, and whether they are *symmetric* or not. We will then discuss subdivided network topologies, and the role of learning. Unless indicated otherwise, the material on complex systems presented below is drawn from [1].

Linear vs. Non-Linear Networks

If the value of nodes is a linear function of the influences from the nodes linked to it, then the system is linear, otherwise it is non-linear. Linear networks have a single *attractor*, i.e. a single configuration of node states that the network converges towards no matter what the starting point, corresponding to the global optimum. This means we can use a 'hill-climbing' approach (where each node always moves directly towards increased local utility) because local utility increases always move the network towards the global optimum.

Non-linear networks, by contrast, are characterized by having multiple attractors and multiple-optima utility functions, like that shown in Figure 2:

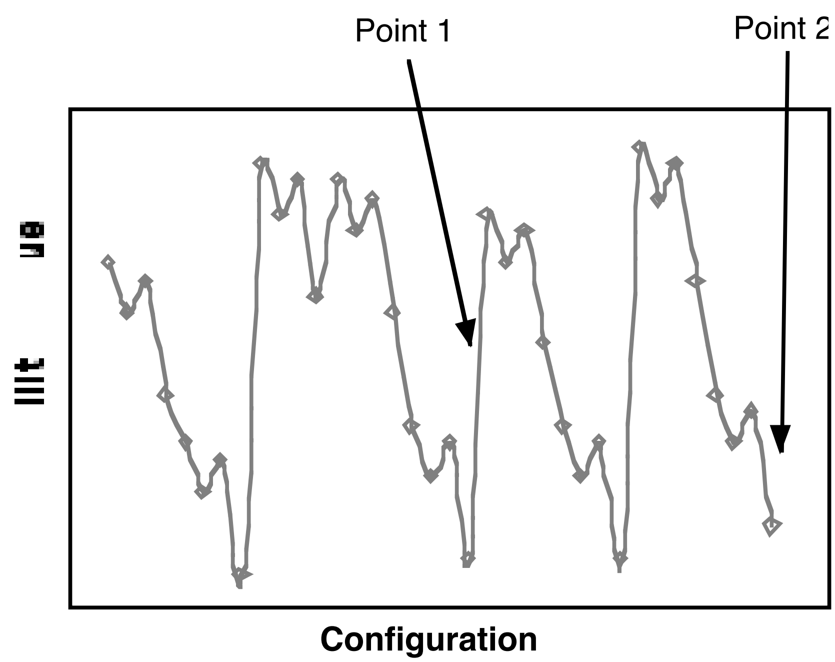

Figure 2. An ultrametric utility function.

A key property of non-linear networks is that search for the global optima can *not* be performed successfully by pure hill-climbing algorithms, because they can get stuck in local optima that are globally sub-optimal. Consider, for example, what would happen if the system started searching at Point 1 in Figure 2 above. Hill-climbing would take it to the top of the local optimum, which is substantially lower than optima in other regions of the utility function. Hill-climbing would do even more poorly if it started at Point 2.

One consequence of this reality is a tendency to stick near well-known designs. When a utility function has widely separated optima, once a satisfactory optimum is found the temptation is to stick to it. This design conservatism is exacerbated by the fact that it is often difficult to compare the utilities for radically different designs. We can expect this effect to be especially prevalent in industries, such as commercial airlines and power plants, which are capital-intensive and risk-averse, since in such contexts the cost of exploring new designs, and the risk of getting it wrong, can be prohibitive.

A range of techniques have emerged that are appropriate for finding optima in ultrametric utility functions, all relying on the ability to search past valleys in the utility function. Simulated annealing, for example, endows the search procedure with a tolerance for moving in the direction of lower utility that varies as a function of a virtual 'temperature'. At first the temperature is high, so the system is as apt to move towards lower utilities as higher ones. This allows it to range widely over the utility function and possibly find new higher peaks. Since higher peaks are also typically wider ones, the system will tend to spend most of its time in the region of high peaks. Over time the temperature decreases, so the algorithm increasingly tends towards pure hill-climbing. While this technique is not provably optimal, it has been shown to get close to optimal results in most cases.

Annealing, however, runs into a dilemma when applied to systems with multiple actors. Let us assume that at least some actors are self-interested 'hill-climbers', concerned only with directly maximizing their local utilities, while others are 'annealers', willing to accept, at least temporarily, lower local utilities in order to increase the utility in other nodes. Simulation reveals that while the presence of annealers always increases *global* utility, annealers always fare *individually* worse than hill-climbers when both are present [2]. The result is that globally beneficial behavior is not individually incented.

How do these insights apply to collaborative design? Linear networks have been used successfully to model *routine* design [3], involving highly familiar requirements and design options, as for example in automobile brake or transmission design [4]. Today's most challenging and important collaborative design problems (e.g. concerning software, biotechnology, or electronic commerce) are, however, *not* instances of routine design. They typically involve *innovative* design, radically new requirements, and unfamiliar design spaces. It is often unclear as a result where to start to achieve a given set of requirements. There may be multiple very different good solutions, and the best solution may be radically different than any that have been tried before. For such cases non-linear networks seem to represent a more accurate model of the collaborative design process.

This has important consequences. Simply instructing each design participant to optimize its own design subspace as much as possible (i.e. 'hill-climbing') can lead to the design process getting stuck in local optima that may be significantly worse than radically different alternatives. Design participants must be willing to explore alternatives that, at least initially, may appear much worse from their individual perspective than alternatives currently on the table. Designers often show greater loyalty to producing a good design for the subsystem they are responsible for, than to conceding to make someone else's job easier, so we need to find solutions for the dilemma identified above concerning the lack of individual incentives for such globally helpful behavior. We will discuss possible solutions in the section below on "How We Can Help".

Symmetric vs. Asymmetric Networks

Symmetric networks are ones in which influences between nodes are mutual (i.e. if node A influences node B by amount X then the reverse is also true), while asymmetric networks do not have this property. Asymmetric networks (with an exception to be discussed below) add the complication of *dynamic* attractors, which means that the network does not converge on a *single* configuration of node states but rather cycles indefinitely around a relatively small *set* of configurations. Let us consider the simplest possible asymmetric network: the 'odd loop' (Figure 3):

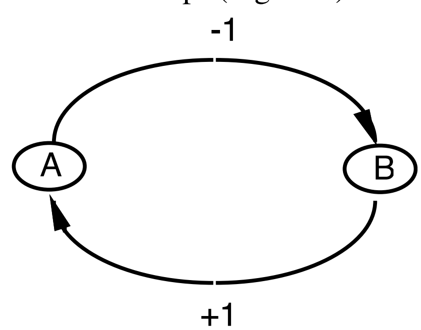

Figure 3. The simplest possible asymmetric network – an 'odd loop'.

This network has two links: one which influences the nodes to have the same value, the other which influences them to have opposite values. Imagine we start with node A having the value 1. This will influence node B to have the value –1, which will in turn influence node A towards the value –1, which will in turn cause node B to flip values again, and so on *ad infinitum*.

Current collaborative design practice is characterized by asymmetric influence loops likely to produce dynamic attractors and therefore non-convergent dynamics. Feedback from later product life cycle perspectives such as manufacturability and transportability, for example. tends to be weaker and slower than influences from design to these perspectives.

Subdivided Networks

Another important property of networks is whether or not they are sub-divided, i.e. whether they consist of sparsely interconnected 'clumps' of highly interconnected nodes. When a network is subdivided, node state changes can occur within a given clump with only minor effects on the other clumps. This has the effect of allowing the network to explore more states more rapidly. This effect is in fact widely exploited in design communities, where it is often known as *modularization*. This involves intentionally creating subdivided networks by dividing the design into subsystems with pre-defined standardized interfaces, so subsystem changes can be made with few or any consequences for the design of the other subsystems. The key to using this approach successfully is defining the design decomposition such that the impact of the subsystem interdependencies on the global utility is relatively low, because the standardized interfaces rarely represent an optimal way of satisfying these dependencies. In most commercial airplanes, for example, the engine and wing subsystems are designed separately, taking advantage of standardized engine mounts to allow the airplanes to use a range of different engines. This is not the optimal way of relating engines and wings, but it is good enough and simplifies the design process considerably. If the engine-wing interdependencies were crucial, for example if standard engine mounts had a drastically negative effect on the airplane's aerodynamics, then the design of these two subsystems would have to be coupled much more closely in order to produce a satisfactory design.

Imprinting

One common technique used to speed network convergence is *imprinting*, wherein the network influences are modified when a successful solution is found in order to facilitate quickly finding (similar) good solutions next time. A common imprinting technique is reinforcement learning, wherein the links representing influences that are satisfied in a successful final configuration of the network are strengthened, and those representing violated influences weakened. The effect of this is to create fewer but higher optima in the utility function, thereby increasing the likelihood of hitting such optima next time.

Imprinting is a crucial part of collaborative design. The configuration of influences between design participants represents a kind of 'social' knowledge that is generally maintained in an implicit and distributed way within design organizations, in the form of individual designer's heuristics about who (i.e. which individual or design group) should talk to whom when about what. When this knowledge is lost, for example due to high personnel turnover in an engineering organization, the ability of that organization to do complex design projects is compromised. It should be noted, however, that imprinting reinforces the tendency we have already noted for organizations in non-linear design regimes to stick to tried-and-true designs, by virtue of making the previously-found optima more prominent in the design utility function.

How Can We Help?

Once the design of a complex artifact has been distributed to many players, encouraging proper influence relationships and local search strategies is the primary tool available to design managers, and should therefore be supported by computer-supported collaborative design technology. This can occur in several ways. Such technology can help monitor the influence relationships between design participants. One could track the volume of design-related exchanges or (a more direct measure of actual influence) the frequency with which design changes proposed by one participant are accepted as is by other participants. This can be helpful in many ways. Highly asymmetric influences could represent an early warning sign of non-convergent dynamics. Detecting a low degree of influence by an important design concern, especially one such as environmental impact that has traditionally been less valued, can help avoid utility problems down the road. A record of the influence relationships in a successful design project can be used to help design similar future projects. Influence statistics can also be used to help avoid repetitions of a failed project. If a late high-impact problem occurred in a subsystem that had a low influence in the design process, this would suggest that the influence relationships should be modified in the future. Note that this has the effect of making a critical class of normally implicit and distributed knowledge more explicit, and therefore more amenable to being preserved over time (e.g. despite changes in personnel) and transferred between projects and even organizations.

Computer-supported collaborative design technology can also help assess the degree to which the design participants are engaged in routine vs innovative design strategies. We could use such systems to estimate for example the number and variance of design alternatives being considered by a given design participant. This is important because, as we have seen, a premature commitment to a routine design strategy that optimizes a given design alternative can cause the design process to miss other alternatives with higher global optima. Tracking the degree of innovative exploration can be used to fine-tune the use of innovation-enhancing interventions such as incentives, competing design teams, introducing new design participants, and so on.

References

[1] 1. Bar-Yam, Y., *Dynamics of complex systems*. 1997, Reading, Mass.: Addison-Wesley. xvi, 848.

[2] 2. Klein, M., et al., *Negotiating Complex Contracts*. 2001, Massachusetts Institute of Technology: Cambridge MA USA.

[3] 3. Brown, D.C. *Making design routine*. in *Proceedings of IFIP TC/WG on Intelligent CAD*. 1989.

[4] 4. Smith, R.P. and S.D. Eppinger, *Identifying controlling features of engineering design iteration*. Management Science, 1997. **43**(3): p. 276-93.