The Dynamics of Collaborative Design: Insights From Complex Systems and Negotiation Research

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

Abstract

Almost all complex artifacts nowadays, including physical artifacts such as airplanes, as well as informational artifacts such as software, organizations, business processes, plans and schedules, are defined via the interaction of many, sometimes thousands of participants, working on different elements of the design. This *collaborative design* process is typically expensive and time-consuming because strong interdependencies between design decisions make it difficult to converge on a single design that satisfies these dependencies and is acceptable to all participants. Recent research from the complex systems and negotiation literatures has much to offer to the understanding of the dynamics of this process. This paper reviews some of these insights and offers suggestions for improving collaborative design.

The Challenge: Collaborative Design Dynamics

Almost all complex artifacts nowadays, including physical artifacts such as airplanes, as well as informational artifacts such as software, organizations, business processes, plans and schedules, are defined via the interaction of many, sometimes thousands of participants, working on different elements of the design. This *collaborative design* process is challenging because strong interdependencies between design decisions make it difficult to converge on a single design that satisfies these dependencies and is acceptable to all participants. Current collaborative design approaches are as a result typically characterized by heavy reliance on expensive and time-consuming processes, poor incorporation of some important design concerns (typically later life-cycle issues such as environmental impact), as well as reduced creativity due to the tendency to incrementally modify known successful designs rather than explore radically different and potentially superior ones.

Research on negotiation focuses on understanding what local behaviors are to be expected from (relatively small numbers of) self-interested agents attempting to come to agreements in the face of interdependencies. Complex systems research compliments this perspective by attempting to understand the global dynamics that emerge as the collective effect of many such local decisions. These two perspectives, when brought together, have we believe much to offer to a understanding of the dynamics of collaborative design. The remainder of this paper is dedicated to exploring some of these insights.

A Model of Collaborative Design

Let us first establish a working definition of 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. A complete design for an artifact includes issues that capture the *requirements* for the artifact, the *specification* of the artifact itself (e.g. the geometry and materials), the *process* for creating the artifact (e.g. the manufacturing process) and so on through the artifacts' entire life cycle. 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.

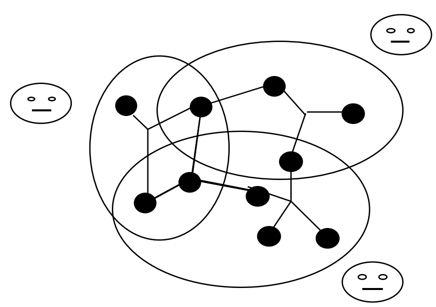

Figure 1: A Model for Collaborative Design

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). Figure 1 below illustrates this model: the small black circles represent design issues, the links between the issues represent design issue inter-dependencies, and the large ovals represent the design *subspace* (i.e. subset of design issues) associated with each design participant. In a large artifact like a commercial jet there may be millions of components and design issues, hundreds to thousands of participants, working on hundreds of distinct design subspaces, all collaborating to produce a complete design.

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 (though in practice we may only be able to assess *comparative* as opposed to *absolute* utility values). A simple utility function might look like the following:

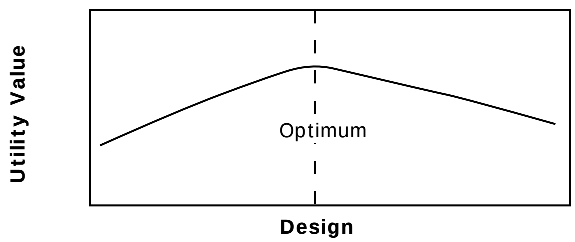

Figure 2. A simple utility function.

The *goal* of the design process can thus be viewed as trying to find the design with the optimal (maximal) utility value, though often optimality is abandoned in favor of ‘good enough’.

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. Improving the efficiency, quality and creativity of the collaborative innovative design process requires, we believe, a much better understanding of the dynamics of such processes and how they can be managed. In the next section we will review of the some key insights that negotiation and complex systems research offers for this purpose.

Insights from Complex Systems and Negotiation 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 maximizes its consistency with the influences from the other nodes. The dynamics of such networks emerge as follows: since all nodes update their local state 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.

The negotiation literature adds the following refinement to this model. Each one of the nodes is *self-interested*, i.e. attempts to maximize its own local utility, at the same time it is seeking a satisfactory level of consistency with the nodes it is inter-dependent with. A central concern of negotiation research is designing the rules of encounter between inter-dependent nodes such that each node is individually incented to make decisions that maximize *social welfare*, i.e. the global utility of the collected set of local decisions. In this case, we can define global utility simply as the sum of node utilities plus the degree to which the inter-node influences are satisfied.

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 tries to maximize the utility of the subsystem it is responsible for, while ensuring its decisions satisfy its dependencies (represented as the links between nodes) with other subsystems. 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 early in the process, the sheer complexity of many design artifacts means that eventually no one person is capable of keeping the whole design in his/her head and assessing/refining its global utility. Centralized control of the design decisions becomes impractical, so the design process is dominated perforce by concurrent subsystem design activities (performed within the nodes) done in parallel with subsystem design consistency checks (assessed by seeing to what extent inter-node influences are satisfied). We will assume, for the purposes of this paper, that individual designers are reasonably effective at optimizing their individual subsystems.

The key factor determining network dynamics is the nature of the influences between nodes. There are two important distinctions: whether the influences are *linear* or not, and whether they are *symmetric* or not. We will consider each one of these distinctions in turn, with an important side trip into the negotiation literature to understand the dilemmas raised by the presence of self-interested agents. This will be followed by a discussion of subdivided network topologies, and the role of learning. Unless indicated otherwise, the material on complex systems presented below is drawn from (Bar-Yam 1997).

Linear vs. Non-Linear Networks

Non-Linearity Produces Multi-Optimum Utility Functions: 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. Their utility function thus looks like that shown in Figure 2 above. 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 utility functions with multiple peaks (i.e. local optima) and multiple attractors, as in Figure 3:

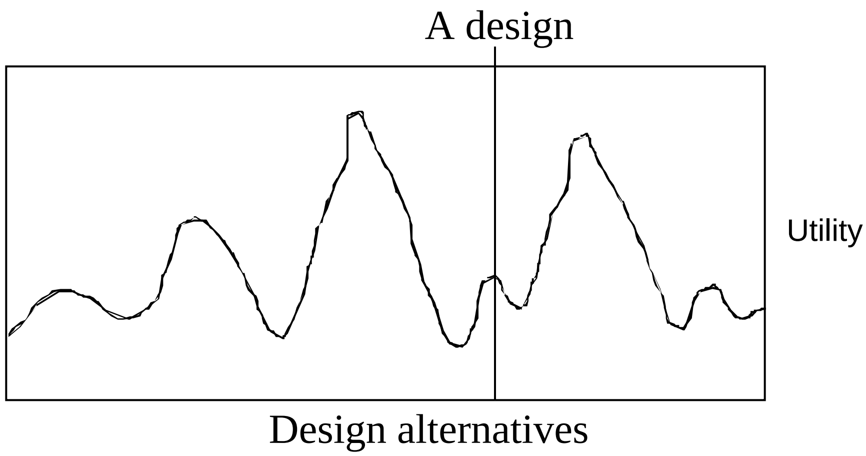

Figure 3. A multiple optima utility function.

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 happened in Figure 3 above. Hill-climbing took the design to the top of a local optimum which has substantially lower utility than some other designs.

To make this concrete, let us examine the following simple example: a network consisting of binary-valued nodes where each node is influenced to have the same value as the nodes it is linked to, and all influences are equally strong (Figure 4):

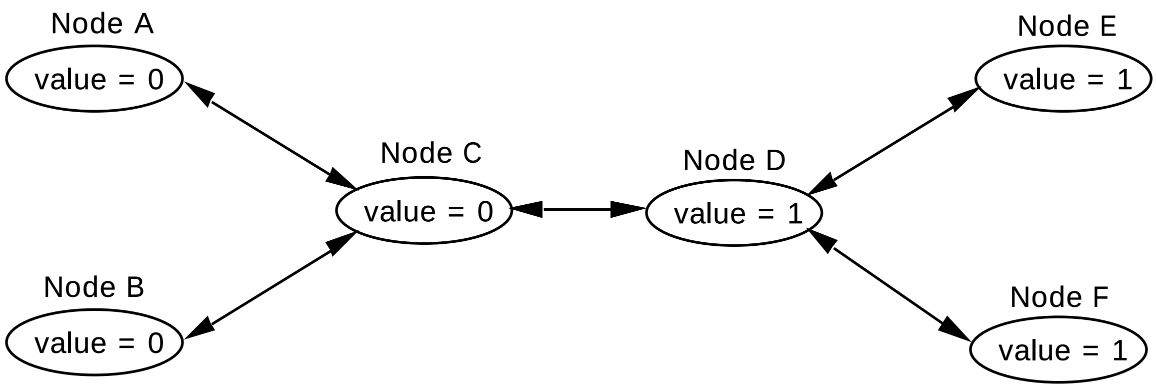

Figure 4: A simple network illustrating how networks can get stuck in local optima.

Node A, for example, is influenced to have the same value as Node C, while Node C is influenced to have the same value as Nodes A, B and D. For simplicity's sake, we assume that the global utility is determined solely by the degree to which the inter-node influences are satisfied. We can imagine using this network to model a real-world situation wherein there are six subsystems being designed, with two equally optimal design options for each, and we want them to use matching interfaces.

This network has reached a stable state, i.e. no single node change will result in an increase in the number of satisfied influences. If we change the value of node A from 0 to 1, it will violate its one influence so this change will not be made. If we change the value of Node C to 1, it will now satisfy the influence with Node D but violate two influences (with Nodes A and B), resulting in a net loss in the number of satisfied influences, so this change will not be made either. The analogous argument applies to all the other nodes in the network. The system will *not* as a result converge on a global optimum (i.e. an ideal design where all the influences are satisfied), even though one does exist (where all nodes have the same value).

A range of techniques have emerged that are appropriate for finding global optima in multi-optima utility functions, all relying on the ability to search past valleys in the utility function. Stochastic approaches such as simulated annealing have proven quite effective (Kirkpatrick, Gelatt et al. 1983). Simulated annealing 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 generally tend to also be wider ones, the system will 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.

A Social Dilemma with Self-Interested Agents: Annealing runs into a dilemma, however, when applied to systems with self-interested agents. Let us assume that at least some actors are ‘hill-climbers’, concerned only with maximizing their local utilities, while others are ‘annealers’, willing to accept, at least temporarily, lower local utilities as part of the exploratory process. We can use a simulation approach to explore what happens. Table 1 summarizes the results for such experiments, giving the local and global utilities achieved for different pairings of agent strategies in simulated non-linear negotiations:

	Agent 2 hill-climbs	Agent 2 anneals
Agent 1 hill-climbs	[.86] .73/.74	[.86] .99/.51
Agent 1 anneals	[.86] .51/.99	[.98] .84/.84

Table 1: Annealing vs hill-climbing agents.

In this table, the cell values are laid out as follows:

[<global optimality>]
<agent 1 optimality >/<agent 2 optimality>

Details on the negotiation results described in this paper are available, unless otherwise specified, in (Klein, Faratin et al. 2002a) and (Klein, Faratin et al. 2002b).

These results show that, while annealers increase *global* utility, and are therefore highly desirable, annealers always fare *individually* worse than hill-climbers when both are present. Hill-climbing is thus a ‘dominant’ strategy: no matter what strategy the other agent uses, it is better to be a hill-climber. If all agents do this, however, then they forego the higher individual utilities they would get if they both annealed. Individual strategic considerations thus drive the system towards the strategy pairing with the *lowest utility values.*

What can be done about this? This pattern of utility values is an instance of a well-known phenomenon in game theory known as the “prisoner’s dilemma” (Osborne and Rubinstein 1994). It has been shown that this dilemma can be avoided if there are repeated interactions between agents (Axelrod 1984). The idea is simple. Each agent uses an annealing strategy at first, but if it determines that the agent it is negotiating with is using hill-climbing, it itself then switches to hill-climbing for its future negotiations with that agent, thereby forcing them both into the 'lose-lose' quadrant of Table 1. It turns out that this 'tit for tat' approach incents annealing behavior in all agents, *assuming* that they negotiate with each other multiple times. This idea can be refined with the addition of a 'reputation mechanism', wherein agents consult a database of previous negotiations (in addition to their individual experience) in order to determine whether the agent they currently face tends to be an annealer or hill-climber. Ideally, however, we would prefer to find a way to incent annealing behavior within the context of a single negotiation, without the requirement of multiple interactions. Can this be done?

Some apparently reasonable approaches are, it turns out, quite ineffective. One approach, for example, is what we can call 'adaptive' annealing. A negotiation typically consists of a relatively large number of offers and counter-offers, resulting in increasingly better interim agreements that eventually are accepted as final by both parties. An agent could therefore in principle switch in mid-stream from being an annealer to being a hill-climber if it determines that the other agent is being a hill-climber. Determining the strategy type of the agent you are negotiating with is in fact relatively easy: an annealer tends to accept a much higher percentage of interim proposals than a hill-climber. The problem with this approach is that determining the type of an agent in this way takes *time*. Our simulations have shown that the divergence in acceptance rates between annealers and hill-climbers only becomes clear *after most of the utility has been committed*, so it is too late to fully recover from the consequences of having started as an annealer if you negotiated with a hill-climber. Hill-climbing therefore remains the dominant strategy. Another possibility is for annealers to simply be less concessionary, i.e. less willing to accept utility-decreasing interim agreements. This in fact allows us to eliminate the poor annealer payoffs that underlie the prisoner's dilemma, but only at the cost of radically reduced global utility. In both cases, we are unable to incent agent strategies that optimize the global utility of the outcome.

Resolving the prisoners' dilemma within the scope of a single negotiation can be achieved, however, through the use of what we call a 'parity-enforcing annealing mediator'. Rather than requiring that the agents anneal, we move the annealing into a third party we call a mediator. In this approach, possible agreements are generated (in our experiments they were generated by the mediator, but this is a not a critical part of the scheme) and then voted on by the negotiating agents. The mediator is a kind of annealer: it is endowed with a time-decreasing willingness to at least temporarily follow up on design proposals that one or both agents voted against. Agents are free to remain hill-climbers in their voting behavior, and thus avoid making harmful concessions. The mediator, by virtue of being willing to provisionally pursue utility-decreasing agreements, can traverse valleys in the agents' utility functions and thereby lead the agents to win-win solutions. Paradoxically, using a mediator that occasionally *ignores* agent preferences leads to outcomes that are better for both agents.

Achieving maximal global utilities in this scheme requires that agents be able to annotate their votes with strength information. A binary scheme is sufficient, wherein agents annotate their accept votes as being either *strong* or *weak*. This allows the possibility of 'over-rides', wherein the mediator pursues an interim agreement that was strongly preferred by one agent and weakly rejected by another. Over-rides are important because such agreements are likely to increase global utility. Agents might of course be tempted to exaggerate in such contexts, marking every vote as being a strong one. But this possibility can be foiled by enforcing running parity on the number of times each agent over-rides the other. This works for the following reason. One can think of this procedure as giving agents 'tokens' that they can use to gain over-rides. A truthful agent spends its tokens exclusively on over-rides that truly offer it a strong local utility increase. An exaggerator, on the other hand, will spend tokens even when the utility increment it derives is relatively small. At the end of the day, the truthful agent has spend its tokens more wisely and to better effect.

Lessons: How do these insights apply to collaborative design? Generally speaking, linear networks represent a special case (only a tiny fraction of all possible influence relationships are linear), but they have proven adequate for modeling what has been called *routine* design. Routine design involves highly familiar requirements and design options, as for example in automobile brake or transmission design (Brown 1989). In these contexts, designers can usually start the design process near enough to the final optimum that the process acts as if it has a single attractor. Previous research on design dynamics has focused on this class of design model, generating such useful results as approaches for identifying design process bottlenecks (Smith and Eppinger 1997) and for fine-tuning the lead times for design subtasks (Eppinger, Nukala et al. 1997).

Rapid technological and other changes have made it increasingly clear, however, that many of the most important collaborative design problems (e.g. concerning software, biotechnology, or electronic commerce) involve *innovative* design, radically new requirements, and unfamiliar design spaces. It is often unclear how 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. One is a tendency to stay with 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 impact of getting it wrong, can be prohibitive.

Another consequence is that collaborative design as currently practiced is probably quite prone to getting stuck in local optima that may be significantly worse than radically different alternatives. Annealing-like processes potentially applicable to addressing this problem are widely used in human collaborative design settings. ‘Brainstorming’, for example, with its emphasis on not pruning candidate solutions too quickly, can be viewed as a kind of annealing. Designers are, however, generally much more strongly encouraged to create a good design for their own subsystems, than to concede to make someone else’s job easier. This incentive structure leads to the “prisoner’s dilemma” described above.

The prisoner’s dilemma can, as we have seen, be avoided if we assume that agents have multiple negotiation encounters and use a ‘tit for tat’ scheme for deciding when to be concessionary or not. Such schemes are probably used, in fact, by many designers in collaborative settings. The relative infrequency of major negotiations, the absence of reputation databases, and high turnover in personnel may, however, sabotage the efficacy of such strategies. It seems likely, in addition, that many engineers make some use of the other approaches we described above, being adaptive or simply highly sparing in how much they concede. These are, after all, apparently reasonable strategies. They do not, however, have the desired result of fostering the discovery of more optimal overall designs. Mediation, as we have seen, has the potential of resolving the prisoner’s dilemma, and it in fact has an important place in current collaborative design practice. Senior engineers, and in some cases teams of such engineers (sometimes called “change boards”) are often called upon to mediate situations where the achievement of satisfactory global utility appears to be threatened. Engineers with that level of experience are, however, a scarce resource, so this tactic is typically reserved for only the most serious problems.

In brief, it appears likely that current collaborative design practice, particularly for highly innovative design, is prone to getting stuck in unnecessarily suboptimal solutions. We will discuss possible solutions to these problems in the section “How We Can Help” below.

Symmetric vs. Asymmetric Networks

Asymmetry Allows Non-Convergence: 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 (if they have cycles in them; see below) add the complication of having *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 cyclic asymmetric network: the ‘odd loop’ (Figure 5):

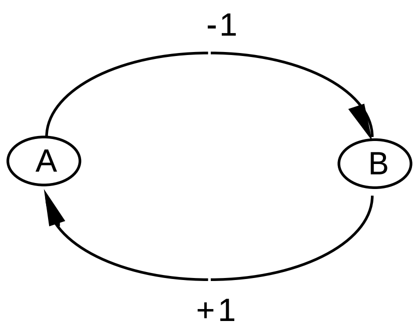

Figure 5. The simplest possible cyclic asymmetric network – an ‘odd loop’.

This network has two links: one where node B influences node A to have the same value, and another where node A influences node B to have the opposite value. Imagine both nodes have the initial value 1, and update each other in parallel. The states of the two nodes will proceed as follows:

State	**Value of Node A**	**Value of Node B**
Initial state	1	1
State 1	1	-1
State 2	-1	-1
State 3	-1	1
State 4	1	1

After one time step (state 1) node A will cause node B to ‘flip’ to –1, and node B will leave node A unchanged. After a second iteration (state 2) node A leaves node B unchanged, but node B causes the value of node A to flip. If we trace this far enough we find that the system returns to its initial state (State 4) and thus will repeat *ad infinitum*. If we plot the state space that results we get the following simple dynamic attractor:

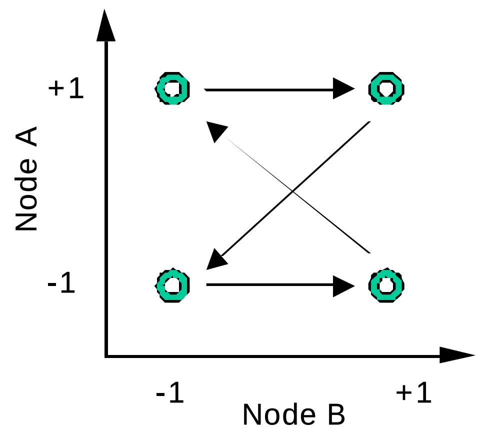

Figure 6. The dynamic attractor for the odd loop.

More complicated asymmetric networks will produce dynamic attractors with more complicated shapes, including ones where states are *never exactly repeated*, but the upshot is the same: the system will not converge. One can always of course stop the system at some arbitrary point along its trajectory, but there is no guarantee that the design utility at that point will be better than that at any other point because the system, unlike the symmetric case, does not necessarily progress monotonically towards higher utility values. This can be understood in the following way. Every utility function can, in principle, be 'compiled' into a (symmetric) network that will progress monotonically towards higher utility values as long as the individual nodes perform local optimization. The opposite, however, is not true. There are many networks (including most asymmetric ones) that do *not* correspond to any well-formed utility function, so their sequences of states clearly can not be viewed as progressing towards a utility optimum (Bar-Yam 1997).

If a network is *acyclic* however (also known as a *feed-forward* network, wherein a node is never able to directly or indirectly influence its own value), it has a well-defined utility function and thus will not have a dynamic attractor.

Lessons: How does this apply in collaborative design settings? Traditional serialized collaborative design is an example of an asymmetric feed-forward network, since the influences all flow uni-directionally from the earlier product life cycle stages (e.g. design) to later ones (e.g. manufacturing) with only weak feedback loops if any. In such contexts the attractors should be static and convergence should always occur, given sufficient time. In such settings we may not, however, expect particularly optimal designs. It is typically very difficult, given the bounded rationality of human beings, for designers earlier in the design life cycle to ensure that the designers later on in the life cycle will be able to produce near-optimal solutions for their very different but highly dependent problems. This is in fact the rational underlying the adoption of concurrent engineering approaches. ‘Pure’ concurrent engineering, where all design disciplines are represented on multi-functional design teams, encourage roughly symmetric influences between the participants and thus can also be expected to have convergent dynamics with static attractors. Current collaborative design practice, however, is a hybrid of these two approaches, and thus is likely to have the combination of asymmetric influences and influence loops that produces dynamic attractors and therefore non-convergent dynamics.

This, moreover, is a fundamental problem. As noted above, it is in principle straightforward to compute the proper inter-node influences given a global utility function. In design practice, however, we do *not* know the global utility function, especially once we have reached the realm of detailed design. The space of possible designs, and the cost of calculating their individual utility values, is simply too large. At best the global utility function is revealed to us incrementally as we generate and compare different candidate designs. The influence relationships between designers are, as a result, invariably defined directly based on experience and our knowledge of design decision dependencies. But such a heuristic approach can easily lead to the creation of influence networks that do not instantiate a well-formed utility function, and thus display dynamic attractors.

Dynamic attractors were found to not to have a significant effect on the dynamics of at least some routine (linear) collaborative design contexts (Eppinger, Nukala et al. 1997), but may prove more significant in innovative (non-linear) collaborative design. It may help explain, for example, why it sometimes takes so many iterations to fully propagate changes in complex designs (Klein 1994).

Subdivided Networks

Subdivision Can Speed Convergence: 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, as for example in Figure 7:

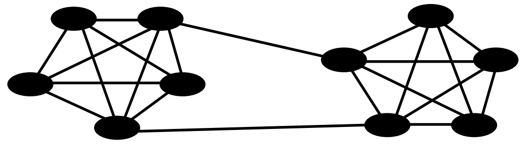

Figure 7. An example of a subdivided network.

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. Rather than having to wait for an entire large network to converge, we can rely instead on the much quicker convergence of a number of smaller networks, each one exploring possibilities that can be placed in differing combinations with the possibilities explored by the other sub-networks (Simon 1996).

Lessons: 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 utility impact of the subsystem interdependencies on the global utility is relatively low, because 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 almost certainly *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

Imprinting Captures Successful Influence Patterns: 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.

Lessons: 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 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, and thus may be counter-indicated for challenges requiring highly innovative designs.

How We Can Help?

What can we do to improve our ability to do innovative collaborative design? We will briefly consider several possibilities suggested by the discussion above.

Information systems are increasingly becoming the medium by which design participants interact, and this fact can be exploited to help monitor the influence relationships between them. 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 previous failed and successful design projects can be used to help better manage future projects. This will require being able to determine which influences were critical in these previous efforts. If a late high-impact problem occurred in a subsystem that had a low influence in the design process, for example, this would suggest that the relevant influence relationships should be modified in the future. Incentive mechanisms can be put in place that reward engineers not just for producing good subsystem designs, but also for participating in what are believed to be productive patterns of mutual influence with other designers. 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, as well as transferred between projects and even organizations.

Information systems can also potentially be used to help assess the degree to which the design participants are engaged in routine (i.e. optimization-driven) vs innovative (i.e. highly exploratory) 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. As with simulated annealing, it will probably make sense to encourage more conceding and exploration early on in the design process, and gradually transition to hill-climbing as time goes on.

The prisoner's dilemma incentive structure that leads to suboptimal designs can be addressed in at least two ways that are probably under-utilized in current practice. One is by the introduction of reputation mechanisms. If we simply make information available on which designers have a history of conceding sparingly, we are likely to find an increase in concessionary behavior, and therefore improved design outcomes, even in the absence of explicit (e.g. salary) incentives. Another possibility is the wider use of mediators. Mediators in collaborative design contexts have traditionally been senior engineers capable of dictating the *content* of a design outcome. Recent work on negotiation algorithms suggests, however, that mediators can be effective by guiding the design *process*, for example as we suggested above by occasionally having the agents follow up on design options that one or both rejected, and by enforcing rough parity in the number of mixed wins. Process-oriented mediation does not require the same depth of domain expertise as content-oriented mediation, and it is therefore likely that designers can be trained to provide that for each other, and that such mediation can become much more widely available as a result.

Finally, information systems can be used to track the history of design alternatives explored and thereby detect the design loops that indicate a non-convergent design process.

Conclusions

Existing collaborative design approaches have yielded solid but incremental design improvements, which has been acceptable because of the relatively slow pace of change in requirements and technologies. Consider for example the last 30 years of development in Boeing's commercial aircraft. While many important advances have certainly been made in such areas as engines, materials and avionics, the basic design concept has changed relatively little (Figure 8):

Figure 8. The Boeing 737 (inaugurated 1965) and the Boeing 777 (1995)

Future radically innovative design challenges, such as high-performance commercial transport, will probably require, however, substantial changes in design processes:

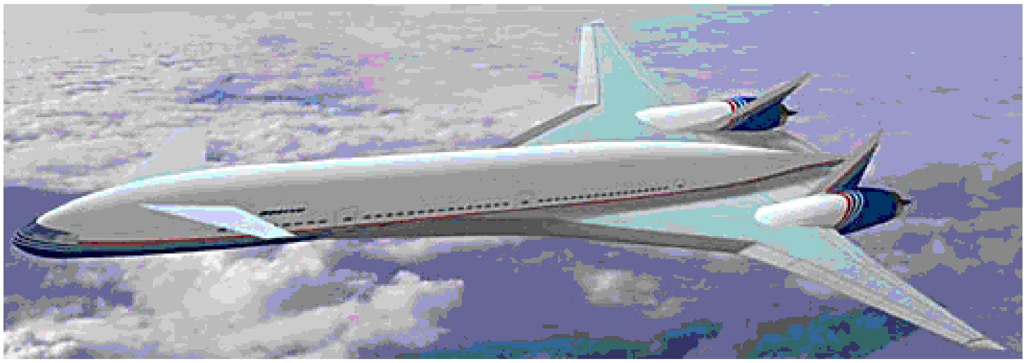

Figure 9. The Boeing Sonic Cruiser (under development)

This paper has begun to identify what recent research on negotiation and complex systems can offer in this regard. The key insights are that important properties of collaborative design dynamics can be understood as reflecting two basic facts: (1) collaborative design is a kind of distributed network, and (2) the agents in this network are self-interested and respond to local incentives. This is powerful because this means that our growing general understanding of networks and negotiation can be applied to help us better understand and eventually better manage collaborative design regardless of the domain (e.g. physical vs informational artifacts) and type of participants (e.g. human vs software-based).

This insight leads to several others. Most prominent is the suggestion that we need to fully embrace an influences- and incentives-centric perspective on how to manage complex collaborative design processes. It is certainly possible for design managers to have a very direct effect on the content of design decisions during preliminary design, when a relatively small number of high-level global utility driven decisions are made top-down by a small number of players. But once the detailed design of a complex artifact has been distributed to many players, the global utility impact of local design changes is too difficult to assess, and design decisions are too voluminous and complex to be made top-down, so the dominant drivers become local utility maximization plus fit between these local design decisions. In this regime encouraging the proper influence relationships and concession strategies becomes the primary tool available to design managers. If these are defined inappropriately, we can end up with designs that take too long to create, do not meet important requirements, and/or miss opportunities for significant utility gains through more creative (far-ranging) exploration of the design space.

References

Axelrod, R. (1984). The Evolution Of Cooperation, Basic Books.

Bar-Yam, Y. (1997). Dynamics of complex systems. Reading, Mass., Addison-Wesley.

Brown, D. C. (1989). Making design routine. Proceedings of IFIP TC/WG on Intelligent CAD.

Eppinger, S. D., M. V. Nukala, et al. (1997). "Generalized Models of Design Iteration Using Signal Flow Graphs." Research in Engineering Design **9**(2): 112-123.

Kirkpatrick, S., C. D. Gelatt, et al. (1983). "Optimization by simulated annealing." Science **220**: 671-680.

Klein, M. (1994). "Computer-Supported Conflict Management in Concurrent Engineering: Introduction to Special Issue." Concurrent Engineering Research and Applications **2**(3).

Klein, M., P. Faratin, et al. (2002a). Using an Annealing Mediator to Solve the Prisoners' Dilemma in the Negotiation of Complex Contracts. Agent-Mediated Electronic Commerce (AMEC-IV) Workshop, Bologna Italy, Springer.

Klein, M., P. Faratin, et al. (2002b). Negotiating Complex Contracts. Autonomous Agents and Multi-Agent Systems, Bologna Italy, AAAI Press.

Osborne, M. J. and A. Rubinstein (1994). A course in game theory. Cambridge, Mass., MIT Press.

Simon, H. A. (1996). The Sciences of the Artificial. Cambridge MA USA, MIT Press.

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