A Complex Systems Perspective on How Agents Can Support Collaborative Design

Mark Klein
Massachusetts Institute of Technology
Peyman Faratin
Massachusetts Institute of Technology
Hiroki Sayama
New England Complex Systems Institute
Yaneer Bar-Yam
New England Complex Systems Institute

1. THE CHALLENGE: COLLABORATIVE DESIGN DYNAMICS

Almost all complex artifacts nowadays, including physical artifacts such as airplanes, as well as informational artifacts such as software, organizational designs, plans and schedules, are created 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.

Complex systems research is devoted to understanding, at a fundamental level, the dynamics of systems made up of interdependent components, and has we argue much to offer to our understanding of the dynamics of collaborative design. Previous research on design dynamics has focused on *routine* design [1] where the design space is well-understood (e.g. as in brake or transmission design), and the goal is to optimize a design via incremental changes for requirements similar to those that have been encountered many times before [2] [3]. 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.

In this paper we will explore some of what complex systems research can contribute to this important challenge. We will begin by defining a simple model of collaborative design, review the strengths and weaknesses of current collaborative design approaches, and discuss some of the insights a complex systems perspective has to offer concerning why it is difficult and what we can do to help.

2. 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. 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.

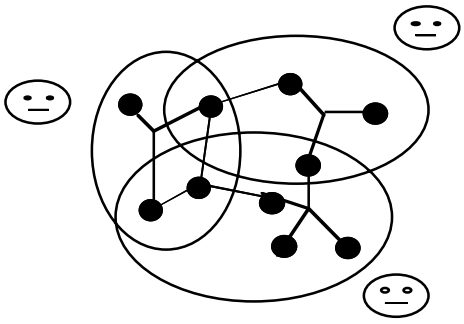

Figure 1: A Model for Collaborative Design

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:

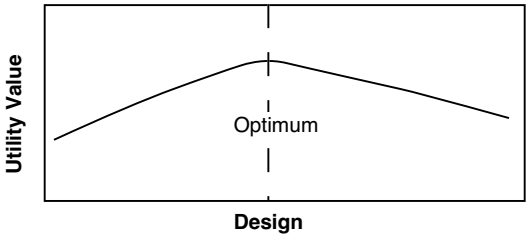

Figure 2. A simple utility function.

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.

3. STRENGTHS AND LIMITATIONS OF CURRENT APPROACHES

Traditionally, collaborative design has been carried out using a serialized process, wherein for example a complete requirement set would be generated, then given to design engineers who would completely specify the product geometry, which in turn would then be given to the manufacturing engineers to create a manufacturing plan, and so on. This has the problem that if an earlier decision turns out to be sub-optimal from the perspective of someone making dependent decisions later on in the design process (e.g. if a requirement is impossible to achieve, or a particular design geometry is very expensive to manufacture): the process of revising the design is slow and expensive, and often only the highest priority changes are made. The result is designs that tend to be poor from the standpoint of later life-cycle perspectives, including for example environmental concerns such as recyclability that are becoming increasingly important.

More recently, several strategies have emerged for better accounting for the interdependencies among collaborative design participants. These include concurrent engineering and least-commitment design:

> *Concurrent engineering* involves the creation of multi-functional design teams, including representatives of all important design perspectives, for each distinct design subspace. Design decisions can be reviewed by all affected design perspectives when they are initially being considered, so bad decisions can be caught and revised relatively quickly and cheaply. While this approach has proven superior in some ways to traditional serial design, it does often incur an overwhelming burden on engineers as they have to attend many hours of design meetings and review hundreds of proposed changes per week [4].
>
> *Least-commitment design* is a complimentary approach that attempts to address the same challenges by allowing engineers to specify a design incompletely, for example as a rough sketch or set of alternatives, and then gradually make the design more specific, for example by pruning some alternatives [5] [6]. This has the advantage that bad design decisions can be eliminated before a lot of effort has been invested in making them fully specific, and engineers are not forced to make arbitrary commitments that lead to needless conflicts.

While the adoption of these approaches has been helpful, major challenges remain. Consider for example the Boeing 767-F redesign program [4]. Some conflicts were not detected until long (days to months) after they had occurred, resulting in wasted design time, design rework, and often even scrapped tools and parts. It was estimated that roughly half of the labor budget

was consumed dealing with changes and rework, and that roughly 25-30% of design decisions had to be changed. Since maintaining scheduled commitments was a priority, design rework often had to be done on a short flow-time basis that typically cost much more (estimates ranged as high as 50 times more) and sometimes resulted in reduced product quality. Conflict cascades that required as many as 15 iterations to finally produce a consistent design were not uncommon for some kinds of design changes. All this in the context of Boeing's industry-leading concurrent engineering practices. The dynamics of current collaborative design processes are thus daunting, and have led to reduced design creativity, a tendency to incrementally modify known successful designs rather than explore radically different and potentially superior ones.

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 can be derived from complex systems research for this purpose.

4. 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 local 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 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 tries to maximize the utility of the design subspace (i.e. subsystem) it is responsible for, while ensuring its decisions satisfy its dependencies (represented as the links between nodes) with other subsystems. As we shall see, to understand network dynamics, the links between nodes need capture only quite abstract properties of the dependencies. 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 checks for consistency among the subsystem designs (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 subsystem utilities.

How do such distributed networks behave? Let us consider 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 3):

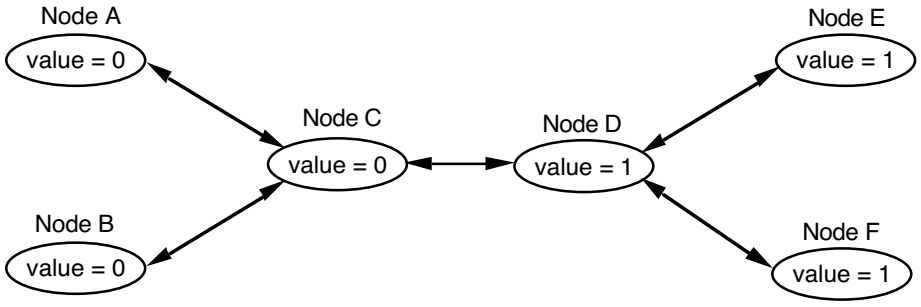

Figure 3: A simple network illustrating how networks can get stuck in local optima.

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. We assume that node values do not effect the overall utility, only 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). 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 first consider how such influences can be defined. We will then consider two important distinctions: whether the influences are *linear* or not, and whether they are *symmetric* or not. We will finally discuss subdivided network topologies, and the role of learning. Unless indicated otherwise, the material on complex systems presented below is drawn from [8].

4.1. How Are Influences Defined?

It is, in principle, straightforward to compute what the inter-node influences should be in order to create a network that implements a given global utility function. In design practice, however, we almost invariably do *not* know the global utility function up front; it is revealed incrementally, rather, by the process of defining and evaluating different candidate designs. Utility evaluations are apt in any case to be approximate at best, because among other things of uncertainties about the context the artifact will exist in. Imagine for example that our goal is to design the most profitable airplane possible: so many imponderable factors heavily influence this (e.g. future oil prices, wars, government subsidies for competitors) that the only way to *really* know the utility of a design is to build it and see what happens! It is usually much easier, as a result, to define the influences directly based on our knowledge of design decision dependencies. We know for example that parts need to have non-overlapping physical geometries, that electrical interfaces for connected systems must be compatible, that weight limits must be met, and so on.

Care must be taken in defining these influences, however. We face the risk of neglecting to give sufficient prominence to important concerns. Traditionally, influences from later stages of the life cycle (e.g. the manufacturing or recycling of the product) tend to be the ones most neglected, and the consequences are only encountered when that life cycle stage has been reached and it is typically much more difficult, time-consuming and expensive to do anything about it. Another concern is that, while there is always a direct mapping from a utility function to a set of influences, the

opposite is *not* true. Asymmetric influences, in particular, do not have a corresponding utility function, and the network they define does not converge to any final result. This will be discussed below further in the section on asymmetric networks.

4.2. 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. 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 4:

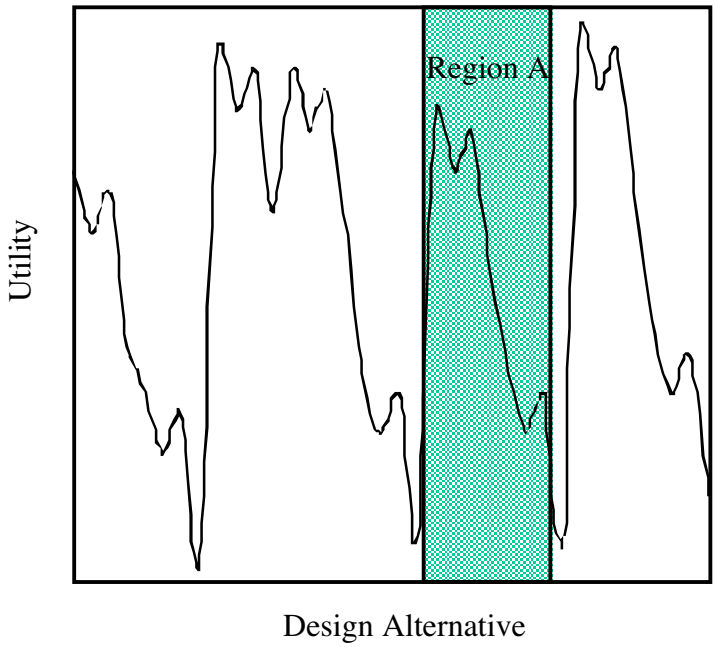

Figure 4. A multiple optima utility function.

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 would happen if the system started searching anywhere in region A in Figure 4 above. Hill-climbing would take it to the top of one of the local optima in this region, all of which substantially lower than some optima outside of region A.

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 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. 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 are also 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 [9]. The result is that globally beneficial behavior is not individually incented.

How do these insights apply to collaborative design? Linear networks represent a special case and we would expect because of this that most collaborative design contexts are non-linear. There is a particular class of collaborative design, however, that has been successfully modeled as linear

networks: *routine* design [1]. Routine design involves highly familiar requirements and design options, as for example in automobile brake or transmission design. Designers can usually start the design process near enough to the final optimum, as a result, to be able to model the design space as having a single attractor. Linear network models of collaborative design have generated many useful results, including approaches for identifying design process bottlenecks [2] and for fine-tuning the lead times for design subtasks [3] in routine design domains.

As we argued above, however, today's most challenging and important collaborative design problems are *not* instances of routine design. The requirements and design options for such *innovative* design challenges are typically relatively unfamiliar, and it is 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".

4.3. 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 5):

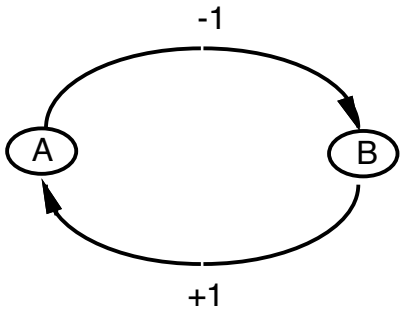

Figure 5. The simplest possible asymmetric network – an 'odd loop'.

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*. If we plot the state space that results we get the following simple dynamic attractor (Figure 6):

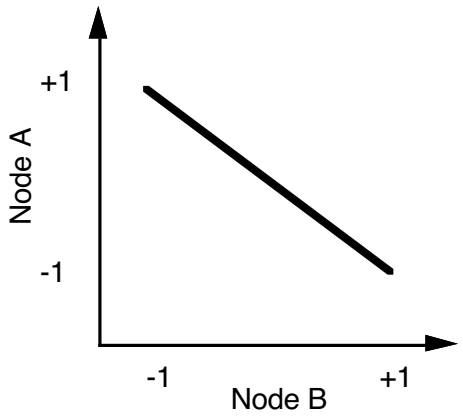

Figure 6. The dynamic attractor for the odd loop.

More complicated asymmetric networks will produce dynamic attractors with more complicated shapes, but the upshot is the same: the only way to get a definite solution (i.e. configuration of node states) with a dynamic attractor is to arbitrarily pick one point along its length. There is one important special case, however: feed-forward networks. The influences in feed-forward networks are *acyclic,* which means that a node never is able to

directly or indirectly influence its own value (there are in other words no *loops*). Feed-forward networks do not have dynamic attractors.

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 at all. In such settings we may not expect particularly optimal designs but the attractors should be static and convergence should always occur, given sufficient time. '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. Dynamic attractors were found to not to have a significant effect on the dynamics of at least some routine (linear) collaborative design contexts [3], 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 account for all changes in complex designs [4].

4.4. 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, as for example in Figure 7:

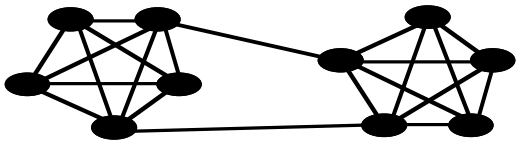

Figure 7. An example of a subdivided network.

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.

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.

4.5. 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 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.

5. HOW CAN AGENTS HELP?

What can we do to improve our ability to do innovative collaborative design? We will consider several possibilities suggested by the discussion above, all oriented around the use of agent-based software systems.

Software 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 a successful design project can be used to help design 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.

Software systems can also potentially be used to 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.

Agent-based systems represent a natural fit for this kind of problem, as it is both large in scale as well as inherently distributed spatially and functionally. Perhaps the most obvious way to use agents is to assign one to each human participant involved in the design process. That agent could monitor the activity of its human ‘client’, infer influence relationships design exploration and concession strategies, cue its client and/or design managers when these influences appear to be diverging from desired values, and provide normative influence relationship information, when turnover occurs, to the new design participants taking on the same role. We can also imagine agents dedicated to monitoring the design activity patterns for smaller and

larger groupings of agents, for example to help detect potentially non-converging design processes.

6. 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 a complex systems perspective can offer in this regard. The key insight is that the dynamics of collaborative design can be understood as reflecting the fundamental properties of a very simple abstraction of that process: distributed networks. This is powerful because this means that our growing understanding of such networks 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 embrace a change in thinking about 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 global utility driven high-level decisions are made top-down by a small number of players. But once the design of a complex artifact has been distributed to many players, the design decisions are too complex to be made top-down, and the dominant drivers become local utility maximization plus fit between these local design decisions. In this regime encouraging the proper influence relationships and local search 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.

Software systems, particularly if configured as agent-based systems to mirror the substantial inherent distributed-ness of the collaborative design process, represent a potentially promising approach for monitoring and even influencing design dynamics.

7. REFERENCES

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

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

[3] Eppinger, S.D., M.V. Nukala, and D.E. Whitney, Generalized Models of Design Iteration Using Signal Flow Graphs. *Research in Engineering Design*, 1997. **9**(2): p. 112-123.

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

[5] Sobek, D.K., A.C. Ward, and J.K. Liker, Toyota's Principles of Set-Based Concurrent Engineering. *Sloan Management Review*, 1999. **40**(2): p. 67-83.

[6] Mitchell, T.M., L.I. Steinberg, and J.S. Shulman, A Knowledge-Based Approach To Design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1985. **PAMI**(7): p. 502-510.

[7] Mezard, M., G. Parisi, and M.A. Virasoro, *Spin glass theory and beyond*. 1987, Singapore ; New Jersey: World Scientific. xiii, 461.

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

[9] Klein, M., et al., Negotiating Complex Contracts. 2002. *Proceedings of the I.nternational Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS-02). AAAI Press.