The Topology of Large-Scale Engineering Problem-Solving Networks

Dan Braha[1, 2] **and Yaneer Bar-Yam**[2, 3]

[1] Faculty of Engineering Sciences, Ben-Gurion University

E-mail: brahad@bgumail.bgu.ac.il,   URL: http://necsi.org/affiliates/braha/dan_braha-Description.htm

[2] New England Complex Systems Institute
E-mail: yaneer@necsi.org,   URL: http://necsi.org/faculty/bar-yam.html

[3] Department of Molecular and Cellular Biology, Harvard University

Distributed problem solving, which often involves an intricate network of interconnected tasks carried out by hundreds of designers, is fundamental to the creation of complex manmade systems. The interdependence between the various tasks makes the system development (referred to as Product development, PD) fundamentally iterative. This process is driven by the repetition (rework) of tasks due to the availability of new information (generated by other tasks) such as changes in input, updates of shared assumptions or the discovery of errors. In such an intricate network of interactions, iterations occur when some development tasks are attempted even though the complete predecessor information is not available or known with certainty. As this missing or uncertain information becomes available, the tasks are repeated to either verify an initial estimate/guess or to come closer to the design specifications. This iterative process proceeds until convergence occurs.

Design iterations, which are the result of the PD network structure, might slow down the PD convergence or have a destabilizing effect on the system's behavior. This will delay the time required for product development, and thus compromise the effectiveness and efficiency of the PD process. For example, it is estimated that iteration costs about one-third of the whole PD time while lost profits result when new products are delayed in development and shipped late. Characterizing the **real-world structure**, and eventually the **dynamics** of complex PD networks, may lead to the development of guidelines for coping with complexity. It would also suggest ways for improving the decision making process, and the search for innovative design solutions.

The last few years have witnessed substantial and dramatic new advances in understanding the large-scale structural properties of many real-world complex networks. The availability of large-scale empirical data and the advance in computing power have led to a series of discoveries that have uncovered statistical properties, which are common to a variety of diverse real-world social, information, biological and technological networks including the world-wide web, the internet, power grids, metabolic and protein networks, food webs, scientific collaboration networks, citation networks, electronic circuits, and software architecture. These studies have shown that many complex networks exhibit the **"small-world"** property of short average path lengths between any two nodes despite being highly clustered. The second property states that complex networks are characterized by an inhomogeneous distribution of nodal degrees (the number of nodes a particular node is connected to) following a power law distribution (termed **"scale free"** networks). Scale-free networks have been shown to be robust to random failures of nodes, but vulnerable to unexpected failure of the highly connected nodes. A variety of network growth processes that might occur on real networks, and that lead to scale-free and small-world networks have been proposed. The third property shows that the system-level structure of complex networks is best approximated by a **hierarchical network organization** with **seamlessly nested modularity**. In contrast to current intuitive views of modularity, which assume the coexistence of relatively independent groups of nodes, real-world networks have an inherent self-similar property: There are many highly integrated small modules, which group into a few larger modules, that in turn can be integrated into even larger modules.

Planning techniques and analytical models that conceive the PD process as a network of interacting components have been proposed before. However, others have not yet addressed the large-scale statistical properties of real-world PD task networks. In the research we report here, we study such networks. We show that task networks from a variety of different large-scale organizations (including software development) have properties (sparseness, small world, scaling regimes, nested modularity) that are like those of other biological, social and technological networks. We demonstrate a previously unreported observation involving an **asymmetry** between the distribution of incoming links (in-degree) and the probability of outgoing links (out-degree). Specifically, the incoming link distributions have sharp cutoffs that are substantially lower than those of the outgoing link distributions (sometimes the outgoing cutoffs are not even present). Analyzing the (Pearson) correlations between in-degrees and out-degrees reveals almost no correlation. Also, tasks with large out-degree generally have small in-degree, and those with large in-degree have small out-degree.

In summary, the study of complex network topologies across many fields of science and technology has become a rapidly advancing area of research in the last few years. One of the key issues is understanding the network properties that are optimized by specific network architectures. We have analyzed the statistical properties of real-world networks of people engaged in product development activities, and have shown that complex PD networks display similar statistical patterns to other real-world networks of different origins. We have addressed the following questions: In the context of product development, what is the meaning of these patterns? How do they come to be what they are? We propose several explanations for these patterns.

The identified statistical properties of large-scale engineering problem-solving networks offer a new perspective and open questions on the **functionality**, **dynamics**, **robustness**, and **fragility** of complex engineered systems. In particular,

1) The “scale-free” property suggests that complex PD task networks are dominated by a few **highly central tasks**. The negative correlation between in- and out-degree for tasks with large total degree implies that, generally, there is a clear distinction between large-scale generators of information (i.e. with high out-degree) and large-scale consumers (i.e. with high in-degree). This further suggests that a distinction has to be made between in- and out-centrality. The ‘failure’ (e.g., excessive rework or lack of integration ability) of central PD tasks will likely affect the vulnerability of the **overall** PD process. Focusing engineering efforts on central PD tasks will likely improve the performance of the PD process. What are the other means, besides the connectivity of each task, by which task centrality can be measured?
2) A variety of network growth processes that might occur on real networks, and that lead to “scale-free” and “small-world” networks have been proposed. What are the implications of these models to the **evolutionary dynamics** of complex engineered systems? Also, is the task network a “mirror image” of the related design network (product architecture)? One can compare their statistical properties.
3) Can the asymmetry between the distribution of incoming and outgoing links be explained by considering the dynamical interactions that take place in distributed problem solving? Might it be related to differences between the actor's capacity to process information provided by others and the actor's capacity to transmit information over the network?
4) PD networks appear to be highly optimized when both PD completion time and product performance are accounted for. Might this suggest that an evolutionary process that incorporates generic optimization mechanisms (e.g., minimizing a weighted sum of development time and product quality losses) leads to the formation of a PD network structure with the “small-world” and “scale-free” properties?
5) Reusing modules at the product architecture level has a direct effect on the task level of product development, and allows firms to reduce the complexity and scope of the product development project by exploiting the knowledge embedded in reused modules. Thus the product development time is significantly reduced. Do the highly connected tasks of the “scale-free” design network tend to be the most reusable modules?
6) What are the implications of the nested modularity characteristic of task networks on the process of design?