Simulating the emergent behavior of complex software-intensive organizations

Alan M. Christie, David A. Fisher
For submission to the ProSim 2000 Workshop

Summary
Organizations are composed of individuals and groups of individuals who interact with their local neighbors. These individuals and groups inherently have limited horizons both with whom they interact and in what information they can access. Because of this limited visibility no one individual can "see" the edges of the universe in which they live and in this sense the system can be considered unbounded. A major characteristic of unbounded systems is the fact that global behavior can emerge which cannot be predicted from local actions. These issues have direct relevance to the understanding of software-intensive organizations. However, performing experimental investigations to assess these organizational properties is virtually impossible. Thus we need a simulation language in which unbounded systems and emergent properties and be simulated and examined. Few, if any existing simulation languages explicitly provide the appropriate mechanisms and this has motivated the development of the Easel language, the topic of this paper.

Background
Recent advances in information technology are having a fundamental effect on the way companies now do business. Speed of communications, rapid access to information, and ease of interaction between different business entities have all been significant influences in this area. The resulting competitive climate has meant: great success for those who can adapt effectively, wrenching changes for others, and has put many out of business [2]. By the nature of their work, software-intensive businesses have been on the leading edge of these changes. Software-intensive companies must deal with dynamically changing conditions (in terms of growth, personnel turnover, and product evolution etc.) and, to an increasing extent, have to succeed in a dynamically changing context.

To address these issues, organizations have tried to "reinvent" themselves through reorganization, downsizing, mergers, changing cultural values etc. However, the consequences of such changes are quite difficult to predict. A significant contributor to this is that changes can result in complex series of interactions that have unforeseen consequences [5].

- To make the above more concrete for software-intensive organizations, the following questions help illustrate the issues[1]:
  What are the consequences of using geographically distributed groups for software development (e.g., using collaborative technology)?
- What is the impact of workflow automation on software development?
- What is the influence of cultural and organizational factors (trust, authoritarianism, customer orientation etc) on software development?
- What is the effect on software development when collaboration between culturally different organizations is involved?
- What is the effect of the development organization's size or structure in successfully completing software projects?
- Can a highly distributed organization produce hierarchically structured software?

---

[1] Note that these examples are illustrative - we do not claim to have addressed them.

These questions imply the need to understand the effects of large numbers of participants, each of which interacts with local participants. Similarly, in these scenarios, participants do not have access to all relevant information, but only to a subset. These are the characteristics of "unbounded" systems, that is, systems in which each individual participant has a restricted view of information in the "world." Thus, decisions that participants make may seem locally appropriate but can turn out to have unintended ramifications when seen globally. The resulting system properties are called emergent and are difficult to predict [1].

We believe that concepts of local action and visibility, unboundedness, and emergent properties are important factors in simulating systems where large numbers of loosely coupled participants are involved. To this end, we are developing a new general-purpose simulation language called Easel, which is currently being implemented [4]. This paper reviews the concepts behind Easel, describes some of its unique properties and summarizes how Easel can be used to simulate organizational behavior in a competitive environment. Let us review some of the conceptual foundations of Easel.

Emergent properties

Emergent properties are analogous to the properties of biological and social systems in that each participant performs simple local actions involving interactions with other participants but without complete knowledge of who else is participating. In these systems, extremely complex global properties emerge from the simple actions and interactions of the participants in the absence of central control or administrative authority. Examples include behavior in ant colonies [3], the culture created by people of a region, and the national economy. Internet examples include the use of chat rooms, the overall governance of the Internet, and the combining of independently developed search tools in ways unanticipated by their authors. An emergent algorithm produces global system-wide properties that result from the collective actions of the participating actors. These global properties may be arbitrarily complex, but result from the interactions of large numbers of individual actors each autonomously performing simple local actions with clearly defined protocols of neighbor interaction.

Actors and neighbors.

Easel's architecture is designed so that it can simulate very large numbers of independent "actors" (analogous to the participants discussed above). Actors are simulated entities of the physical world (e.g., a system administrator, user, intruder, automobile, bird, or the moon), of the electronic world (e.g., a computer, router, or peripheral device), or of the software world (e.g., a software agent or task). Each actor has it own thread of control and this allows for a high degree of parallelism in Easel's execution. Actors can interact directly only with their near neighbors and only in ways prescribed by their neighbor relationships. Neighbor relationships are protocols of interaction and are defined as types that can be associated with any actor. Thus, in a simulation of birds in flight, a bird's near neighbors might be any bird or other object that the bird can see from its current position and heading. In an organizational simulation, an actor's near neighbors might be only those actors that are connected by formal organizational ties. In the latter example, neighbor operations might include sending and receiving messages.

Local visibility and unboundedness

Actors are easily constrained to emulate the limited visibility of the real world they describe. The finite horizon that can be experienced by each actor results in system properties that cannot be determined by any individual actor (in a sense actors cannot see the wood for the trees). Thus to any actor the system appears to be unbounded and the emergent system-wide properties that arise cannot be predicted solely from local information. In other words, unbounded networks may be characterized by distributed administrative control without central authority, by limited visibility beyond the boundaries of local administration, and for any individual actor, by incomplete information about the network's topology and component functions.

Conclusions

In software-intensive organizations, planning decisions are made on the basis of currently available information that is inherently incomplete and imprecise. Such decisions can be consistent with the available information and its implications, but may result in unexpected consequences in the broader, more informed context of the real world in which they are applied. Specifically, in dynamic, rapidly evolving, and stressful environments, as exemplified by software-intensive organizations, actors

- attempt to reach their goals with imperfect or biased information and with severe time constraints
- have to deal with information that, while possibly accurate, may be insufficiently complete to expose critical issues
- interact with neighboring actors, with varying degrees of trustworthiness when exchanging information.

may respond to different actors, or even the same actors, in different ways (e.g., as competitors or partners).

The primary purpose of the Easel language is to simulate unbounded systems. One such type of system is reflected by software-intensive organizations that, to succeed, are forced to rapidly adapt to a dynamically changing external environment. Key requirements for Easel (language and software system) include: an execution semantics consistent with unbounded networks, a one-to-one mapping between the simulation description and what is being simulated, the ability to support a broad spectrum of applications, and support for monitoring, data collection and analyses within a simulation. Another key requirement is an easy-to-use depiction facility that will help analysts visualize emergent properties and algorithms. The Easel language is now specified and a significant portion of the implementation is complete. We are currently working on several applications problems in the areas of computer network survivability, organizational dynamics, neural networks, and infrastructure assurance.

It is hoped that this work will ultimately spark interest in using Easel within the software simulation community particularly in the context of simulating organizational behavior in competitive environments.

References

1. Cottrel, A, Emergent Properties of Complex Systems, in The Encyclopedia of Ignorance - Everything you ever wanted to know about the unknown. pp 129-135, Premagon Press, 1977.

2. Davidow W. H., Malone, M. S., The Virtual Corporation, HarperCollinns Books, 1992.

3. Dorigo, M., The Ant System: Optimization by a colony of Cooperating Agents, IEEE Trans. on Systems, man and Cybernetics, Part-B Vol. 26, No 1 pp. 1-13, 1996 (see also http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html).

4. Fisher, D. A., and Lipson, H. F.,. Emergent Algorithms - A New Method for Enhancing Survivability in Unbounded Systems. Proceedings of the Hawaii International Conference On System Sciences, January 5-8, 1999, Maui, Hawaii.

5. Tenner, E., Why things bite back - technology and the revenge of unintended consequences. Knopf Publishing, 1996.