

An Agent-Based Computational Model of a Self-Organizing Project Management Paradigm for Research Teams

Paul Robinette*, John Seiffert*, Ryan Meuth*, Ryanne Dolan**, Donald Wunsch*

*Applied Computational Intelligence Laboratory, Missouri University of Science and Technology

**University of Missouri

Abstract

We propose a new research organization management paradigm to increase throughput of projects by allowing researchers to choose their own projects [through self-organization](#). [Our methods draw upon the field of Agent-Based computational social science where Artificial Life and simulated societies have been used to study complex systems including economies and financial markets](#). [Modeling the researchers as individual agents, we simulate our new management structure](#) against a more traditional organization where the researchers are broken into departments based on their skills and assigned projects by management. [Our results, measuring the amount of time it takes a research organization to serve a given number of contracts, show promise in the less hierarchical approach](#).

Introduction

Traditional business organizations assign employees to tasks with a management-heavy top-down approach. A direction will be identified by an executive, turned into a high-level set of deliverables by a department head, and [then](#) refined into work packages by several layers of middle management. The last middle manager typically assigns some of his employees to the project and picks one of them to be a project manager. This process requires considerable overhead in the form of managers and administrators and requires managers several layers removed from the actual project to make implementation decisions that affect employees not directly under their control.

Some attempts have been made to break these traditional practices by giving project managers and employees more control. These efforts have mostly concentrated on applying agile software development practices to project management. According to the [Agile Manifesto¹](#), agile practices focus on individuals, customers and adapting to change. Typical methods give developers more control by allowing them to pick functions they would like to implement rather than [relying on managers to decide](#) which programmer is best for which job. These practices were created by software developers who decided software development is considerably different from other forms of engineering and thus required a considerable change in management [structure](#).**

Research organizations are considerably different from both software development and traditional engineering organizations. Researchers are typically comprised of engineers and scientists who are much more self-directed than typical engineers in a production-driven environment. Unfortunately, most research organizations are run by large corporations [that attempt to use the same management systems which prove successful for production-driven departments to organize the work of the research departments](#). This [management approach](#) results in researchers being assigned to projects where they have little interest or little skill and [on which](#) their performance suffers [in comparison to their potential productivity level working on a project better suited to their abilities](#).

We propose a new paradigm for organizing researchers that involves self-organization principles from adaptive systems. Each researcher is assumed to be an intelligent individual who knows his [or her](#) own skills and has enough motivation to find projects of [personal](#) interest. Projects will be identified by senior management and customers. These projects will be posted as a list of deliverables with a reward value for completing the project, much like a bounty on a contract. Researchers will scan the available projects and sign up for the one that they feel fits them the best in terms of skills required and individual interest in the subject. One researcher will be required to volunteer to be the project manager. He [or she, with input from the other researchers on the team](#), will put together a project plan and submit it to senior management [for review and approval](#). This process completely eliminates the need for layers of middle management for project creation and review.

Background

Agile Software Development Practices Applied to Project Management

Agile software development methods encourage a collaborative development environment in which each worker is seen as an intelligent agent capable of making decisions and communicating important information [2](#). Each worker follows a set of rules (depending on the particular strategy employed) that operate at various timescales: immediate peer-review of code, continuous testing of software modules, daily meetings, and bi-weekly deadlines, for example. These rules are designed to increase feedback [among](#) workers, management, and customers, promoting continuous adaptation at every level of the continuously-evolving project. Rather than requiring a global re-design of the software and reorganization of the team each time a new feature or requirement is presented by a customer, agile methods are designed to accommodate changing requirements quickly by isolating the effects to small teams making local and immediate decisions.

Agile methods are regarded as radically different than traditional top-down design and management methods. This was a significant hurdle preventing their early adoption by skeptical managers whose experience and training generally suggested that progress requires order and strict [centralized control](#) [3](#). However, software development is itself radically different than other fields and requires a nontraditional approach in general. Early software development failures led to the exploration of alternative approaches, and agile methods today are widely adopted. In particular, the inability of top-down management and traditional engineering design methods to cope with changing software requirements and fast-paced development cycles signaled a need for something new. These problems arise naturally due to the intrinsic complexity of modern software systems and the difficulty of designing them beforehand.

By adopting adaptive rules and encouraging feedback, agile methods alleviate the burden of an exhaustive initial design and subsequent re-designs. This paper introduces a similar strategy for the management of research teams which is designed to increase productivity when management might not know the optimal allocation of researchers. In both cases, the top-down management approach is rejected in favor of self-organizing teams of intelligent workers making local strategic decisions.

Adaptive Resonance Theory

Adaptive Resonance Theory (ART) was developed by Grossberg [\[art ref\]](#) as a solution to the plasticity and stability dilemma [which asks](#) how adaptable (plastic) should a learning system be so that it does not suffer from catastrophic forgetting of previously-learned rules (stability). [Computational implementations](#) of ART can learn arbitrary input patterns in a stable, fast, and self-organizing way, thus overcoming the effect of learning instability that plagues many other competitive networks [4567](#).

The basic ART [unsupervised neural network](#) architecture consists of two-layers of nodes, the feature representation field F1, and the category representation field F2, as shown in Figure 1. The nodes in layer F1 are activated by the input pattern, while the prototypes of the formed clusters are stored in layer F2. The nodes in layer F2 that are already being used as representations of input patterns are said to be committed. Correspondingly, the uncommitted node encodes no input patterns. The two layers are connected via adaptive weights, W_j , emanating from node j in layer F2. After layer F2 is activated according to the winner-take-all competition [among](#) a certain number of committed nodes and one uncommitted node, an expectation is reflected in layer F1 and compared with the input pattern. The orienting subsystem with the pre-specified vigilance parameter ρ ($0 \leq \rho \leq 1$) determines whether the expectation and the input pattern are closely matched. If the match meets the vigilance criterion, [then learning occurs](#) and the weights are updated. This procedure is called resonance, which suggests the name of ART. On the other hand, if the vigilance criterion is not met, a reset signal is sent back to layer F2 to shut off the current winning neuron, which will remain disabled for the entire duration of the presentation of this input pattern, and a new competition is performed among the remaining nodes. This new expectation is then projected into layer F1, and this process repeats until the vigilance criterion is met. In the case that an uncommitted node is selected for coding, a new uncommitted node is created to represent a potential new cluster.

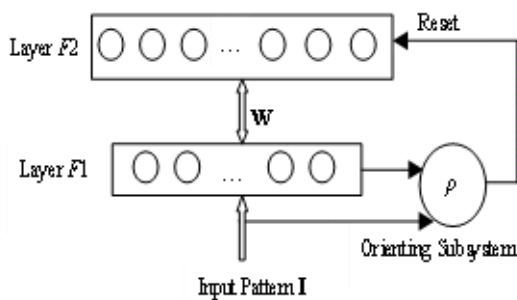


Figure 1 -

$$T(j) = \frac{|x \wedge w^j|}{|w^j|}$$

Equation 1. - ART Category Match

$$\rho \geq \frac{|x \wedge w^j|}{|x|}$$

Equation 2. - ART Vigilance Match

In equations 1 and 2, the norm being taken is the l1 norm and the meet operation represents the Fuzzy AND logic operator and is implemented by taking the element-by-element minimum of the two vectors. In simpler terms, the category match phase of the ART process measures the [proportion](#) of the input vector that is captured by the template,

known as the "bottom-up" activation. Similarly, the vigilance match stage is a measure of how well the template is captured by the input vector, known as "top-down" activation. A best match, or "resonance," is said to occur when [the top-down activation satisfies the vigilance criterion](#).

This bottom-up / top-down resonance property allows ART to exhibit fast, stable, and transparent learning and atypical pattern detection. This property can be applied to project management by [representing](#) the projects as templates and the employees as input vectors. Instead of measuring percentage match, the system would measure probability of success and begin contracts with the highest probabilities. For management to perform an ART-like process, they would have to post contracts, much like in the self organizing organization idea above, and rely on employees to pick projects in a [bottom-up](#) fashion. Management would also assign employees in a [top-down](#) fashion, much like in a traditional organization. Resonance occurs when management agrees that an employee has the intersection of skills and desire to successfully execute a project from the business point of view. Only projects that have sufficient manpower are pursued. This idea is something of a combination of the self organizing research organization idea and a traditional organization.

Computational Social Science

While traditional methods in the study of complex systems rely on mathematics to [analyze](#) governing dynamics, the emerging field of computational social science [Miller] is based on a generative multi-agent modeling framework capable of demonstrating emergent phenomena [which are difficult or impossible to capture using](#) the classical differential equation approaches [Epstein]. The approach we take in this paper to investigate project management techniques falls under the purview of this new school.

Agent-based computational economics [Testafasion] pioneered the use of computer simulations in the social sciences, and it is to this field that we mostly look for inspiration in our project management analysis. Computational economists have studied financial markets [Chiarella], international exchange [Aherns, Arifovic], and can [analyze](#) systems which are particularly difficult to model [mathematically, such as](#) those with heterogeneous agents [Meyer.] Also, these artificial societies have been used in modeling the sustainable development of metropolitan systems [Wang]. It is a natural step to apply computational intelligence methods, such as those from self-organizing neural networks and approximate dynamic programming [Si], which has itself also found application in the modeling of socio-economic systems [Aviv], to this field of computational social science. Our construction of a self-organizing managerial structure is a contribution to this area of research and builds upon the foundation laid by the field of agent-based computational modeling of organizations [Levitt]

Methodology

As our management model is designed to optimize the [productivity](#) of a research organization, we design our agents to be the researchers themselves. Each agent is represented as a vector of several values, each corresponding to the researcher's skill in a category of impact for the organization, such as software, electrical, computer, mechanical and aerospace engineering. We will investigate the organization of these agents in two business structures: a traditional hierarchy and an office built on a philosophy of self-organization.

[Productivity](#) is measured by the successful completion of research contracts. A contract is represented as a vector of six elements, one corresponding to each of the researchers's five skills and one measuring the duration or difficulty of the project. In order for a team of researchers to make progress on a contract the sum of the skills possessed by the agents must cross the threshold indicated in the contract's vector representation.

We initialize a population of researchers with skill levels distributed uniformly from a low value of 0 to a high value of 10. In the traditional office, the researchers are divided into departments based on specialty. The highest-rated agents in each skill are assigned to the appropriate department as equitably as possible, with an even amount of researchers in each department. The reality is, however, that some departments will have staff with non-ideal ratings in the desired skill, just as in real life. Contracts will be assigned to departments based on the highest skill requirement in a winner-take-all fashion. For example, a contract with highest rating in electrical engineering will be assigned to the electrical engineering department. The departmental managers, represented in our model by an optimization algorithm instead of as individual agents, will select researchers for a project team to service the contract in a way that accounts for the contract's skill requirements. The manager selects which researcher to place on the contract by finding the researcher who reduces the total amount of missing skill points [by the largest amount](#). This is accomplished by taking the sum of the differences between the contract's needed skills and the researcher's provided skills. If, for example, a contract is generated with requirements of (15, 20, 10, 25, 5) then the manager will pick a researcher with the skills (10, 8, 2, 9, 2) over a researcher with the skills (5, 2, 4, 3, 10). As a result of dividing the researchers into departments and not permitting the departments to work together, the number of researchers assigned to a team will be relatively high since

reaching the required levels for the skills not specialized in by the department may require a large number of low-skilled individuals. We will see that in the self-organizing management structure this inefficiency is not as likely to be necessary.

For the self-organizing office, the manager algorithm is not used. Instead, each agent chooses which of the available contracts to pursue. This decision is based on a comparison of the contract's requirements with the agent's skill levels, the idea being that an agent will desire to work on a contract which is best suited for the researcher's own skill set. Available contracts are posted on a bulletin board and the most senior agents pick first. If all of the contracts on the board are filled and there are still more researchers in need of jobs, then another contract is posted on the board. In the beginning, this gives little choice to junior researchers. For example, if an available contract with the profile (15, 20, 10, 25, 5) is presented to an agent with skill set (5, 10, 8, 7, 3), then the agent would desire that project since the most important skill for it is the one in which the researcher excels. A contract requiring different skills, such as (20, 2, 10, 10, 20), would be less attractive to that particular agent. The agent decides which contract to select by taking the sum of the differences between his skills and the skills the contract still needs. A contract cannot proceed until it has all of the required skills.

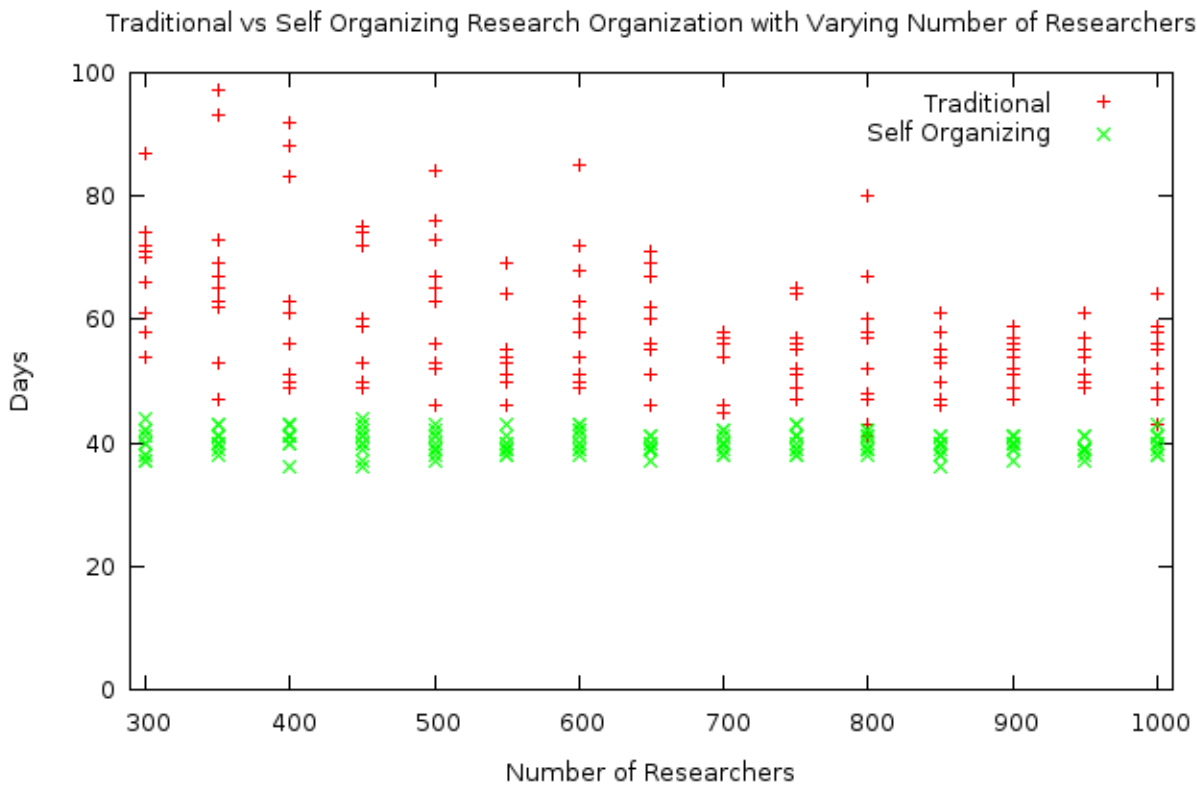
The performance of each management organization—the traditional and the self-organizing—is measured by the amount of time it takes to complete a specified number of contracts.

Results

Using the basic structure outlined above, we run several simulations. We vary the following factors: number of researchers, <whatever else you did>.

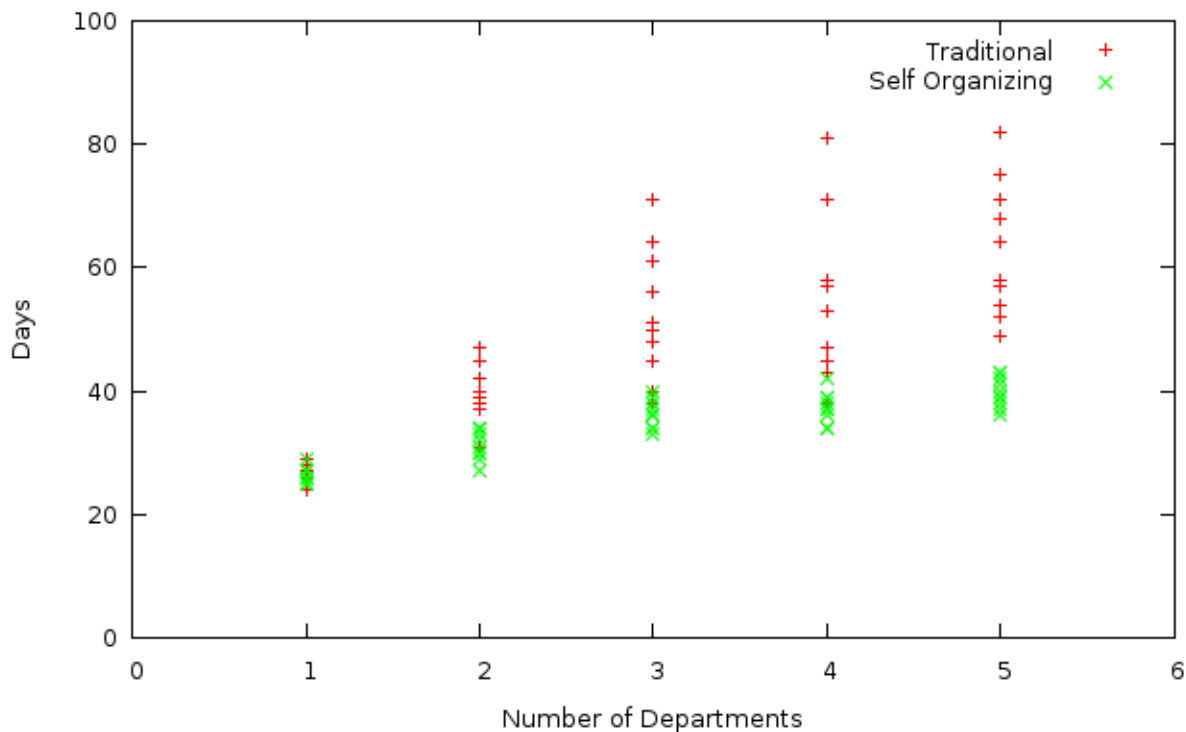
Varying Number of Researchers

For the first simulation, the number of researchers was varied to determine if the self organizing algorithm was better than the traditional algorithm and to ensure that the algorithm was appropriate for several different sized organizations. The number of contracts was set as one third the number of researchers in each case to give the same time range to complete each contract. The number of contracts on the bulletin board at one time for the self organizing company was 25.



The second simulation held the number of researchers at 300 and the number of contracts at 100, but it varied the number of skills measured (and thus the number of departments for the traditional organization) from one to five. The number of contracts on the bulletin board at one time for the self organizing organization was 25.

Traditional vs Self Organizing Research Organization with Varying Number of Departments



Discussion

In general, the self-organizing research organization performed much better than the traditional organization. When the number of researchers was changed the average of the results was approximately 50-100% better, depending on which category was checked. The self-organizing company was very stable through this test, which was to be expected since the number of contracts was directly related to the number of researchers. There is a slight trend in the traditional research organization results that led to faster completion times as the number of researchers increased. This is likely due to a greater mix of skills available in each department.

There is a clear relationship between number of departments and time required for the traditional organization. When there was only one department the traditional organization performed as well as the self organizing one but as the number of departments increased, the time for the traditional organization to complete the task increased. When only one skill was required the decision processes for the traditional and self organizing organizations are remarkably similar, it is just the order of execution that is different. The traditional organization tries to find the best person for the next available contract out of the whole group of researchers while the self organizing organization tries to find the best contract for the next person in the list. The gap between the two organizations widened considerably as the traditional organization was divided into more departments. It is interesting to note that the self organizing organization performed slightly worse on average as the number of skills required increased. This was because it is easy to find the exact skills required when only one skill is being measured but much harder as more combinations of skills are required.

Conclusion

In simulation, the self organizing research organization performed as well as or better than the traditional organization. This lead increased as more skills caused more departments to be created in the traditional organization. The results show good promise for the idea that researchers are more productive when allowed to choose their own contract and pave the way for more in-depth simulations in the future. [We also see much promise in using agent-based computational models in the analysis of project management techniques and organizational structures in general.](#)

Acknowledgments

Support from the National Science Foundation, the Missouri University of Science & Technology Intelligent Systems Center, the Missouri University of Science and Technology Chancellor's Fellowship, and the M.K. Finley Missouri Endowment, are gratefully acknowledged.

References

Ahrens, R., & Reitz, S. (2005) Heterogeneous expectations in the foreign exchange market: Evidence from daily DM/US dollar exchange rates." *Journal of Evolutionary Economics*. 15. 65-82.

Arifovic, J. (1996). The behavior of the exchange rate in the genetic algorithm and experimental economies. *Journal of Political Economy*. 104, 510-541.

Aviv, Y., & Pazgal, A. (2005). A Partially Observable Markov Decision Process for Dynamic Pricing. *Management Science*. 51(9) 1400-1416.

Chiarella, C., Gallegati, M., Leombruni, R., & Palestrini, A. (2003) Asset Price Dynamics among Heterogeneous Interacting Agents. *Computational Economics*, Oct-Dec, 22, 213-223.

Epstein, Joshua. *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press. 2007.

Levitt, Raymond. (2004). Computational modeling of organizations comes of age. *Computational and Mathematical Organization Theory*. 10(2), pp. 127-145.

Meyer, D., Karatzoglou, A., Leisch, F., Buchta, C., & Hornik, K. (2003). A Simulation Framework for Heterogeneous Agents. *Computational Economics*. Oct-Dec, (22).

Miller, John, and Page, Scott. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton Studies in Complexity, Princeton University Press. 2007.

Si, J., Barto, A., Powell, W., & Wunsch, D. (2004). *Handbook of Learning and Approximate Dynamic Programming*. IEEE Press Series on Computational Intelligence. 2004.

Testafasion, L., & Judd, K. (2006) *Handbook of Computational Economics: Agent Based Computational Economics*. North-Holland. Amsterdam, The Netherlands.

Wang, F., & Tang, S. (2004). Artificial societies for integrated and sustainable development of metropolitan systems. *IEEE Intelligent Systems* 19(4), pp. 82-87.

[art ref] [Grossberg, Stephen \(1976\). Adaptive pattern classification and universal recoding, II: Feedback, expectation,](#)

1. Beck et al. Manifesto for agile software development, Dec 2008.
<http://agilemanifesto.org/>.
2. @conference{abrahamsson2003nda,
title={{New Directions on Agile Methods: A Comparative Analysis}},
author={Abrahamsson, P. and Warsta, J. and Siponen, M.T. and Ronkainen, J.},
booktitle={International Conference on Software Engineering: Proceedings of the 25 th International Conference on Software Engineering: Portland, Oregon},
volume={3},
number={10},
pages={244--254},
year={2003}
}
3. @article{karlstrom2005cam,
title={{Combining Agile Methods with Stage-Gate Project Management}},
author={Karlstr{\o}m, D. and Runeson, P.},
journal={IEEE SOFTWARE},
pages={43--49},
year={2005},
publisher={IEEE Computer Society}
}
4. [1] G. A. Carpenter and S. Grossberg, "Fuzzy ART: Fast Stable Learning and Categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759-771, 1991.
5. [2] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks*, pp. 493-504.
6. [3] G. A. Carpenter and N. Markuzon, "ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases," *Neural Networks*, vol. 11, pp. 323-336, 1998.
7. [4] S. Grossberg and G. A. Carpenter, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54-115, 1987.