The Simulation of Idea Propagation in Organizations

Christopher A. White

Bell Laboratories
Alcatel-Lucent Technologies
600 Mountain Ave. Room 2B-417, Murray Hill, NJ 07974
(908) 582-3200
whitec@alcatel-lucent.com

Abstract

The healthy exchange of ideas within an organization leads to faster problem solving, mitigates short and long term risk, and opens the possibility for disruptive technological change. We introduce a new tool (GYRUS) for the simulation and optimization of idea propagation within an organization. This tool treats the organizational topology, internal processes, and implements an individual knowledge model to examine idea propagation. The topology represents both the formal and informal networks of idea movement within an organization. The processes include all activities resulting in the exchange or introduction of ideas to the organization. The knowledge model concerns how individuals store and propagate ideas. We apply this tool to a simple organizational topology to understand the propagation characteristics of ideas and the coupling of ideas between entities in the structure.

1.0 Introduction

The generation and capture of ideas is a critical first step in the innovation process. Typically both individuals and organizations excel at idea generation. Well-tested collaborative brainstorming [1] and proposal generation techniques can generate large numbers of ideas either on a focused topic or across a broad array of disciplines. Beyond the generation step, the propagation of ideas across an organization and the capture of essential inventions which an R&D organization can shepherd into innovations is a significantly more complicated and consequently more important problem. A crucial question in today's environment of ever increasing pressure to decrease time-to-market and push for ever shorter product cycle times concerns how to optimize organizational structures and processes to facilitate idea propagation.

Arguably, the greatest benefit from the healthy exchange of ideas within an organization arises from an increased ability to capture disruptive ideas. Within the context of this paper, we define a disruptive idea as an idea which has not previously been experienced by any member of the organization. In addition to being a new idea, we require a disruptive idea to be a new idea *unrelated* to the vast array of ideas experienced by the organization. It is important to notice that the determination of disruptive is based upon the current state of knowledge of the organization. A disruptive idea for one organization may not be disruptive for another organization. Examples exist throughout the scientific and technical literature where an idea "borrowed" from one discipline enabled rapid progress when applied in a different context in another discipline.

The generation of a disruptive idea, while critically important, represents only the first step in a long process requiring significant effort which actually functionalizes the idea as the solution to a technical challenge, the basis of a new product, or even the opening to an entirely different way of thinking. Within this paper, we use the term idea propagation for the post generation, pre-implementation phase of this process. During idea propagation, the idea is transferred between individuals. The idea is enhanced through this exchange, by combination with similar ideas, by testing against prior work, and by seeding the generation of other new ideas. It is this involved process which eventually determines the extent to which this idea is accepted by the organization and advanced to the functionalization phase.

One focus of recent idea related efforts in industry concerns the rampant interest concerning collaborative brainstorming, or methods of capturing both internal and external ideas. These efforts range from creating lists of critical problems [2], to enabling public voting and input, to putting forth a variety of calls for proposals from both government [3,4] and industry [5]. Bell Labs recently conducted an internal brainstorming effort termed the Grand Challenge *Challenge* (GCC) which attempted to bring together the diverse backgrounds, knowledge, and problem solving methods drawn from the world-wide Bell Labs research organization to select a Grand Challenge research project. The activity included a balance of both bottoms-up brainstorming and top-down guidance to access the "wisdom of crowds" in a manner which led to a well-defined set of focused project proposals. Understanding the movement of ideas within processes such as these is critical for ensuring disruptive ideas are captured and that the ideas reinforced by the process are not just the ideas of the "loudest" individual, but are those which are collectively most important.

The central feature of idea propagation concerns the sharing and communication of ideas. Therefore it is no surprise the internet and a vast array of idea sharing tools are currently being deployed to facilitate the generation and propagation of ideas. These deployments concern commercial endeavors where individuals provide feature requests and criticisms (Apple, Microsoft, etc), enthusiast endeavors where groups of like minded individuals develop the tools and products (Linux, GNU tools, etc.) to satisfy the goals of the group, to purely social conversations between individuals (Facebook, etc.) The ability to focus and guide this global idea factory on path solving the problems of the world is technically within our grasp. However to understand and optimize such a system requires significantly more understanding of idea propagation.

There is a long history of literature related to the adoption of innovations, or the diffusion of innovations [6-10]. Typically, these models relate to the creation of tangible assets and are driven primarily by economic criteria. The long path and significant effort required to commercialize an idea as well as our focus on disruptive ideas which by definition occur rarely, indicate that many of these economic or diffusion based treatments will not provide adequate detail for use in organizational optimization. Fortunately, this prior work does provide a wealth of information for benchmarking our agent based simulation efforts.

We propose to treat the problem of understanding idea propagation through the development of a discrete, agent-based simulation engine. The development of this tool builds on our recent creation of dynamic, hierarchical simulation tools for the study of bandwidth and user behavior in IPTV (IP based television) deployments as well as our development of the ATOM optical network simulation tool [11] for the generation and optimization of power control strategies for optical mesh networks. The model and tool we present here is adaptable to treating a wide variety of different organizations. This could be universities, industrial laboratories, or even an online social community.

The purpose of this research effort is to develop and apply a simulation tool known as GYRUS to explore a variety of organizational topologies, processes, and idea propagation models. This short paper will focus primarily on a simple propagation model and structure of our simulation tool (Section 2) and provide some initial results (Section 3) for some simple organizational structures.

2.0 Model and Simulation tool

We focus on the three core aspects of idea propagation: the organizational topology, the activities or processes within which members of the organization participate, and the knowledge model that mimics how an individual might "remember" or "propagate" an idea. The organizational topology is a representation of both the formal and informal networks of individuals within the organization. The processes concern activities that introduce ideas to the organization and activities that exchange ideas between members of the organization. The knowledge model represents how ideas are stored and propagated by members of the organization. Each of these aspects will be described in more detail below.

The GYRUS simulator is designed in a modular fashion and is capable of adjusting the fidelity of simulation for each of these aspects. For example, it is possible to construct a very complicated knowledge model and to implement only a simple organizational structure or set of processes. Likewise, it is possible to provide a detailed model of the organizational topology, but apply a very simple knowledge model. In this manner, it is possible to focus the computational effort of the simulator on the primary area of interest. This separation further allows independent optimization and verification of the models for each of the aspects and facilitates the reuse of existing assets and the rapid prototyping of new models. The use of python as the implementation language for the simulator allows for very rapid prototyping as well as exhibiting a simple interface with compiled languages for computationally intensive activities.

GYRUS is similar to a series of tools we have developed in several other areas which include large hierarchical systems with complex dynamical behavior. In this case as well as our previous efforts, the complex system dynamics arises not from the creation of a complicated model seeking to capture every detail of the system, but from the simulation of a large number of independent agents presenting simple behavior in parallel.

2.1 Organizational Topology

The most basic computational element within GYRUS is the entity. Within each entity is a list of sub-entities, a list of processes, and a local knowledge model. The list of sub-entities allows an entity to represent a single individual (an atomic entity) or a collection of individuals (a composite entity). For example a department entity might contain a list of all the individual entities who are members of that department. There are no restrictions on the composition of these lists, so the simulator structure allows the representation of an organizational structure where a single individual is a member of multiple formal and informal groups. Further, since atomic and composite entities are functionally equivalent, it is possible to create a nested hierarchy of entities, i.e. a department entity might contain a list of group entities each of which contains a list of individuals.

Each entity also contains a list of processes. Processes represent an activity which introduces or exchanges ideas within the organization. Each entity can execute multiple processes concurrently indicating the possibility of multiple paths for each entity to interact with the other entities both inside and outside the organization. Composite entities contain processes which apply to all sub-entities. For example a department meeting process would be contained within a department entity and might trigger events within all of the department members. The simulated organization sits within a "thermodynamic bath" of random external ideas which can be drawn into the organization through an appropriate set of processes.

The final component of an entity is a local knowledge model. This knowledge model represents the ideas experienced by this entity. Composite and atomic entities both contain knowledge models. In this manner, a composite entity can experience and store an idea in a manner which is different than the storage in the constituent atomic entities. Learning by a composite entity (such as a department) in the simulator is a rough manner of representing a group knowledge which is not necessarily given as the sum of the individual knowledge of the members of the group.

2.2 Process

Processes represent the activities that introduce ideas to the organization, or cause the exchange of ideas within the organization. Typically processes are periodic events that facilitate information exchange. This might be attending a conference, writing a paper, reading a paper, attending an internal seminar, or just spending time to think about a subject. In addition to these simple periodic processes, it is possible to implement significantly more complicated idea triggering processes. For example the introduction of a novel idea might cause an entity to spend more time exploring similar ideas. To encompass this wide range of possible processes, each process is represented as a finite state machine.

For each time step of the simulator, each entity will loop over all processes in its internal process list and advance the state of this process. The initial state of each process is selected randomly to remove the possibility of the synchronization of initial states between different entities causing simulation artifacts in the dynamics. For each time step, all processes advance in parallel for all entities of the system. In practice, for computational efficiency, only those processes that will change state actually consume computational resources. For example, idle or waiting entities only require computational effort when a change of state is expected based on the evolution of time. This allows the GYRUS tool to simultaneously treat very fine time scale activities over short time periods while at the same time allowing simulations to run over a much longer time scale for the full system.

2.3 Knowledge Model

The knowledge model represents the most complicated aspect of simulating the propagation of ideas. Each entity contains a local knowledge model that must be capable of learning an idea, forgetting an idea, propagating an idea, and querying an idea. Within the limited context of this paper, we will discuss a simple vector based knowledge model. In this case, ideas are represented as high dimensional (N=5000) vectors with a normalized magnitude of 1.0. The inner product between two idea vectors represents the degree of similarity, so two collinear vectors represent the same idea, two orthogonal vectors represent completely different ideas. This vector model clearly contains several deficiencies, but it is simple to implement and captures a significant number of the important properties of real ideas. As such, it represents an ideal starting point for building more sophisticated knowledge models. More than this, it represents a simple, well understood model for use when exploring more sophisticated structures and processes.

In addition to the vector, several other quantities are stored within each idea. Each idea contains a representation of the origin of the idea, i.e. was the idea drawn from an external source, or a trusted internal source. Closely related to the origin of the idea is the type of the idea. While the origin reflects the level of confidence in the source, the type reflects the level of confidence in the mechanism of transmission. For the simple models we describe here, neither origin nor type is used in determining the rate of learning for an idea. However, more sophisticated knowledge models exist which enable the acceptance of an idea to be modulated by both origin and type. The implementation allows the easy incorporation of these and other additional parameters for more complex models.

2.3.1 Learning

Learning is the most important function of the knowledge model. Specifically, when an entity experiences an idea (effectively receives a message from another entity containing the idea) some aspect of this idea must be stored by the knowledge model. Typically this storage is not accomplished with perfect fidelity, may include some level of randomness, and perhaps most importantly will depend on the prior knowledge of the entity. For example an individual might more easily learn ideas that are closely related to ideas they already know. Two individuals might experience exactly the same idea, however, what gets stored in the knowledge model will be different. In addition to a dependence on prior knowledge, the origin of the idea might also have an impact on learning. An idea from a trusted source might be stored with a much higher probability than an idea expressed by a stranger.

Our simple vector model satisfies these properties. When an entity experiences an idea, the idea vector is partitioned into a portion that lies within the subspace spanned by all previously learned vectors and a portion that lies outside this subspace. The magnitude of each of these vectors is then multiplied by a scale factor derived from the origin of the idea. Each of these numbers is then compared to a random number selected from 0 to 1.0. If the computed number is larger than the random number, the vector is stored. In the case of the vector inside the subspace, this is accomplished by incrementing a weight for each vector stored previously which comprise the new vector.

In the case of the vector outside the subspace, a new vector is added to the list of stored vectors and its weight is set to 1.0.

In this manner, new ideas are added and the weights of old ideas are reinforced through the introduction of new ideas. The selection of weights as they relate to the origin of the idea might be entity dependent. For the purposes of this paper, we have used equal weights from all sources; however it is not difficult to select a reasonable set of weights based upon the process generating an idea. For example, for some people reading a journal article they might be much more likely to "learn" a new concept than if they were just listening to a seminar.

2.3.2 Forgetting

Just as important as learning, the knowledge model must be capable of forgetting an idea. Much of the dynamics that we expect to find through simulation concerns a balance of learning, reinforcing ideas against forgetting these same ideas. An idea without reinforcement will eventually be forgotten. Within the simple vector model, forgetting is accomplished by a periodic process that decrements the idea weights for each entity. As a first model, the amount of decrement depends on the current weight resulting in an exponential "forgetting" process. Once the idea weight reaches a certain minimal threshold, the idea is removed from the entity. As with learning, it is easy to envision significantly more sophisticated functional forms for forgetting.

2.3.3 Propagating

In addition to the storage aspect of the knowledge model, the presentation of an idea to another individual must be treated. In this case, an entity must be capable of generating an idea, drawn from the set of ideas previously experienced by that entity to present to others. Within the simple vector model, the propagation of knowledge utilizes the list of learned idea vectors and the current list of idea weights. The propagated idea is formed as a sum of each stored the vector multiplied by the idea weight and a random number 0 to 1.0. After accumulating these values, the magnitude of the new propagated idea is again normalized to 1.0. This process produces an idea which will on average most closely resemble the vectors which have the largest weights within the entity.

2.3.4 Querying

To accomplish our goal of measuring in quantitative manner idea propagation, we require the ability to query an entity with an idea and to determine, in this case with a single number, if the entity understands the idea. The query value in the simple vector model is determined as a sum of idea weights multiplied by the square root of the inner products of the idea with each of the stored idea vectors. In this manner, we effectively decompose the idea as a linear combination of the stored ideas, and then return the weighted sum of the coefficients.

3.0 Results

We will present results from two distinct sets of simulations with the GYRUS tool. The first focuses primarily on our simple idea model and its application to a system containing one or two individuals. The second focuses on a more complicated organizational structure containing 20 individuals. In both of these cases, the simulation introduces a disruptive idea at a specific entity of the organization; we will call this the "injected idea". The simulator then queries the other entities of the organization at each time step of the simulation. As output, we obtain as a function of time, the familiarity (or the idea magnitude) of each entity (including the entity where the original idea was introduced) with the injected idea.

3.1 Individuals

Figure 1 shows the result of injecting an idea to a single individual which has no active processes. As would be expected, the injected idea decays as a function of time until the idea is forgotten. For this simulation a threshold of 0.1 has been set as the minimum idea weight, ideas with weights below 0.1 are removed from the storage of the individual and will not be propagated to others.

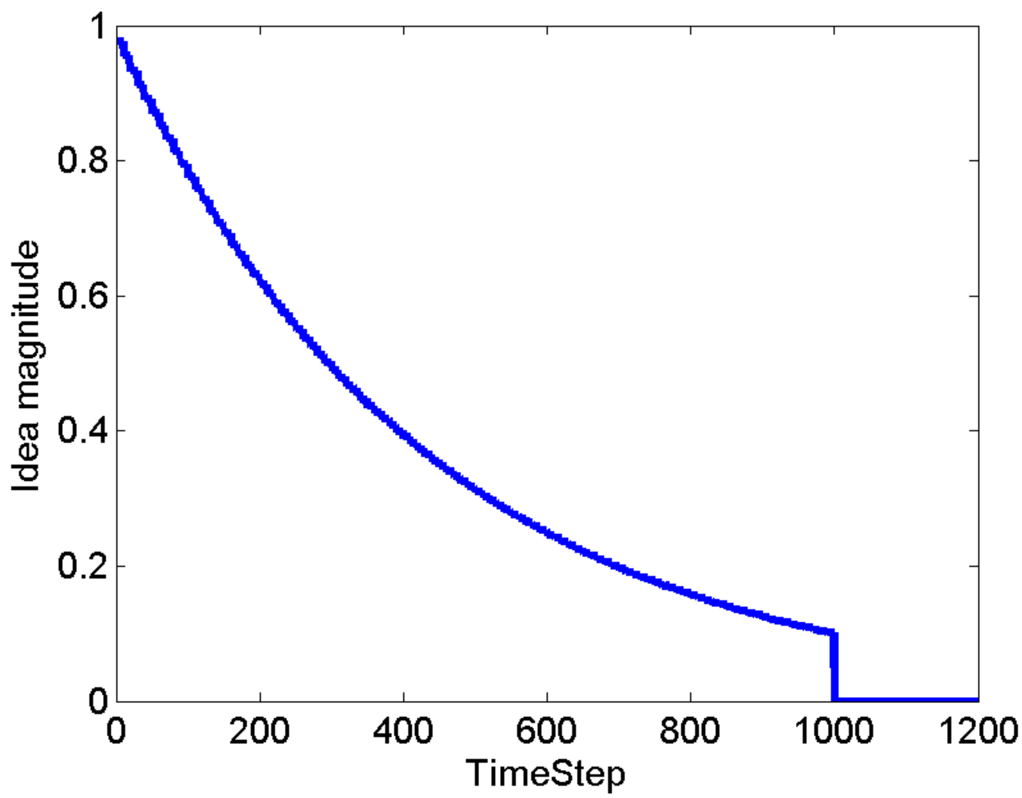

Figure 1: Plot of the idea magnitude for an injected idea at the entity where the idea was injected. The entity contains no reinforcing processes, thus the idea decays to the forgetting threshold (0.1) and the idea is removed from the entity.

Figure 2 show the result of adding a periodic reinforcing process. For these simulations, we have added a second individual. The two individuals interact and potentially exchange ideas every 200 time steps. Within this system, it is possible for the idea to persist beyond the exponential forgetting decay period. The solid line represents a system which includes only the injected idea plus a small amount of noise introduced at the forgetting threshold (0.1). Notice for this artificial system, it is not possible for the idea to be forgotten permanently as it is being periodically reintroduced to the individuals on a time scale faster than ideas can be forgotten. The dotted line, however represents the same system of two individuals. In addition to the idea exchange, we introduce a new random idea to each individual every 200 time steps. As the individuals gain more ideas to propagate, the chances of propagating the injected idea decreases. In this case, as shown in the figure, it is possible for the system to lose knowledge of the injected idea.

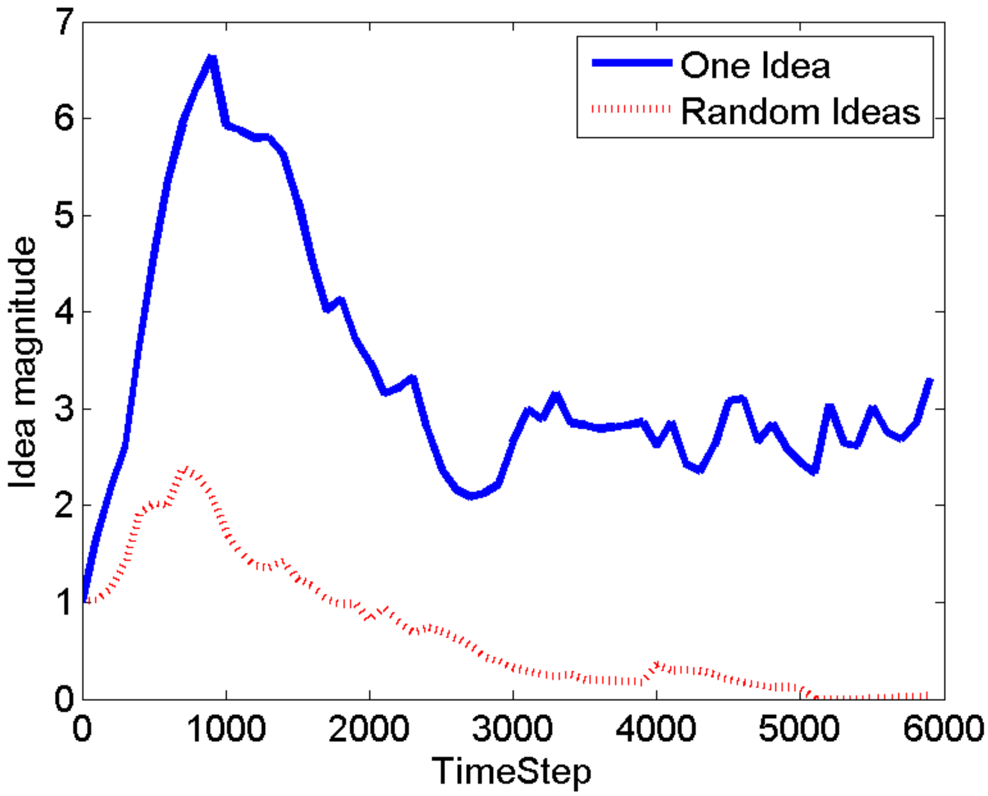

Figure 2: Plot of the idea magnitude for an injected idea at the entity where the idea was injected. The system contains two entities which mutually reinforce the injected idea. The solid line shows the system with only one idea present, since the periodic reinforcing process happens faster than the idea decay process, the idea will not be forgotten. The dotted line shows a similar system with the addition of a periodic injection of random ideas. In this case, the injected idea competes with all ideas in the system for reinforcing and eventually decays below the forgetting threshold.

The situation in figure 2 is interesting because it directly emphasizes the balance between the rate of periodic reinforcement and the rate of forgetting. If the rate of periodic reinforcement is too little, the injected idea will be lost. In the case of the dotted line this occurs as the idea competes for reinforcement with the other ideas in the system. This situation also exposes one deficiency of our current model, if the periodic reinforcement is too quick, the weights for the idea will grow without bound. Essentially the individuals will reinforce at a rate too great in comparison with their rate of forgetting. Simple cases such as these can be used to adjust the parameters (i.e. rate of forgetting) for reasonable values for use within a more complicated organizational structure. Further it is possible to ensure global stability a maximum value of knowledge of an idea by restricting the increase in idea weight possible through reinforcing processes.

3.2 Larger organizations

Figure 3 shows results from a larger organizational structure containing a total of 25 entities, a single department containing four groups each group contains five individuals. Individuals one and two are in group one. Individual three is in group three. The idea was injected at individual one. In addition to the random injection of ideas every 100 time steps, the department exchanges ideas between all individuals every 400 time steps and each group exchanges ideas within the group every 100 time steps.

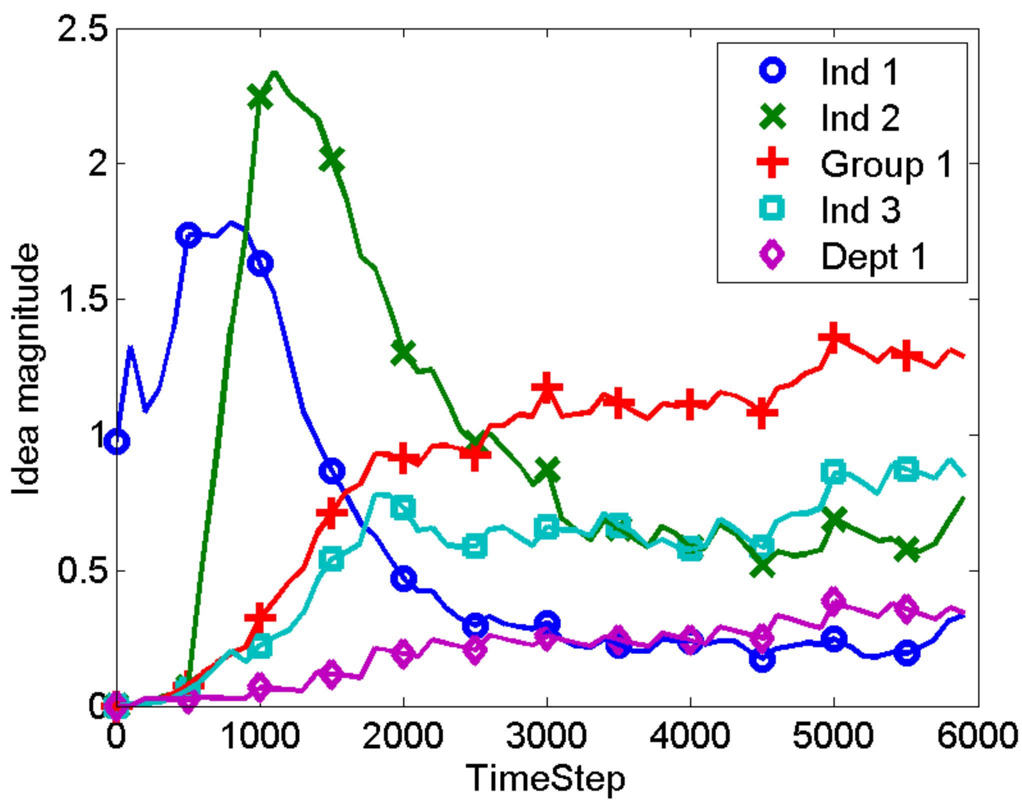

Figure 3: Plot of the idea magnitude for an injected idea at individual one. The organization contained 26 entities, a single department containing four groups each of which contained five individuals. Individuals one and two are in Group one. Individual three is in Group three. The department contained a single meeting process which coupled all individuals which occurs every 400 time steps. Each group contained a meeting process which occurred every 100 timesteps.

There are several interesting features in figure 3. First, we see that by time step 1000, the entity most familiar with our injected idea is individual two. In this case, the idea has been transferred through the periodic group meetings of group one from individual one to individual two. The group one line in the figure also shows an effective transfer of the idea to the group entity. The increase in the idea magnitude of the group is slower but once obtained also decays in a relatively slow manner. Individual three also encounters the idea at later time steps; in this case, the only way for individual three to experience the idea is through a global department meeting. Also transferred during the global department meeting, one can see the slow growth of the idea at the department level.

This figure shows clear windows for capturing this idea at the individual, group and department levels. Clearly the portion of the curves where the idea magnitudes peak can be adjusted through a careful selection of process timings. In addition the development of additional structure that might couple individuals in different groups with a more regular periodic meeting might enable longer period of idea capture.

4.0 Conclusions

The simulation and modeling of idea propagation within an organization is an important component of optimizing the overall organizational operation. We have presented a simple model of idea propagation which treats three specific aspects of the problem. The first aspect relates to the organizational topology and attempts to capture the formal and informal connections within the organization. The second aspect concerns modeling the processes an organization implements to encourage the sharing of ideas. The third aspect relates to the individual's knowledge model which expresses how an individual stores, processes, and propagates ideas. We have implemented this model within the GYRUS simulation tool and applied the tool to several simple test cases. In the future, we expect to explore and apply increasing levels of sophistication in the models as we seek to compare and apply our simulated results to optimize organizational behavior.

References

[1] Stahl, G. Koschmann, T, Suthers, D., Computer-supported collaborative learning: An historical perspective, Cambridge handbook of the learning sciences (2006).

[2] National Academy of Engineering, Grand Challenges for Engineering, http://www.engineeringchallenges.org/ (2008).

[3] NSF Grand Challenge program http://www.nsfgrandchallenge.org/ (2009).

[4] DARPA Urban Challenge program http://www.darpa.mil/grandchallenge/index.asp (2009)

[5] HP Labs Innovation Research Program, http://www.hpl.hp.com/open_innovation/irp/ (2009).

[6] Rogers, E., *Diffusion of Innovation* (Free Press, New York, 1995).

[7] Bettencourt, L.M.A, et.al., The power of a good idea: Quantitative modeling of the spread of ideas from epidemiological models, *Physica A: Statistical Mechanics and its Applications*, Vol. 364, pp. 513-536. (2006)

[8] Hagerstrand, T., *Innovation diffusion as a spatial process*, University of Chicago Press, Chicago (1967).

[9] Heap, S.P.H, Parikh, A., The diffusion of ideas in the academy: A quantitative illustration from Economics, Research Policy, Vol. 34, pp. 1619-1632. (2005).

[10] Sarafoglou, N., Paelinck, J.H.P. On diffusion of Ideas in the Academic World: The Case of Spatial Econometrics, Annals of Regional Science, 42(2), 487-500 (2008).

[11] Chekuri, C., Claisse, P., Essiambre, R., Fortune,S. , Kilper, D.C., Lee, W., Nithi, N.K., Saniee, I., Shepherd, B., White, C.A., Wilfong, G., Zhang, L., *Bell Labs Technical Journal*, 11(2), 129-143.