University of Tennessee Health Science Center
UTHSC Digital Commons

Faculty Publications

Graduate School of Medicine – Knoxville

2009

Understanding Generalist Function

F. Matthew Mihelic

Follow this and additional works at: https://dc.uthsc.edu/gsmk_facpubs

Part of the Policy Design, Analysis, and Evaluation Commons

White Paper
Understanding Generalist Function
F. Matthew Mihelic, MD
July 9, 2009

A major concern in dealing with multi-agent complex adaptive systems is the understanding of how such systems build order in the midst of an otherwise chaotic universe. The ordering principles of such systems that are found at the biological level can be applied to better understanding human organizations, and can also be used to develop or modify highly complex intelligent agent software systems for greater efficacy. This paper describes the theoretical mechanism of how huge multi-agent complex adaptive systems overcome the exponential increases in entropy that would otherwise cause system collapse, through the use of agents that function to maintain the multidimensional architecture of the system concept as defined by the source code of the system. This paper also explores some of the implications and applications that follow from the understanding of generalist function.

Generalist Function

Biological systems are multi-agent complex adaptive systems that build and maintain order despite vast complexity at multiple levels and constantly changing environmental circumstances. There is a heretofore unrecognized level of biological organization that serves to decrease the disorder or entropy that is inherent in the complexity of such multi-agent systems, and this level of biological organization is built and maintained by what can be called generalist agents. Multi-agent complex adaptive systems are systems that are made up of multiple agents that individually possess shared source code, and that work together to perform a function in the face of system perturbations. The source code within each agent can be read linearly, but can be reconfigured multi-dimensionally, and it is the multidimensional reconfiguration of the source code that effects the adaptation of the system. Agents that individually contain an identical source code differentiate from each other by the geometric configuration of the source code which can block or open specific portions of that source code. While differentiation or specialization of agents is necessary for optimally efficient performance of certain functions within the system defined by the source code, it is necessary that some agents maintain an unblocked source code in order to coordinate and integrate the functions of the various types of specialty agents toward the higher level concept of the system source code. This can be done by generalist agents because the open and unblocked source code of such agents contains less entropy than the partially blocked source code of a specialty agent. Stem cells function in an organism as such generalist agents, and the function of generalist agents is necessary for the growth and development of any multi-agent complex adaptive system.

A Theoretical Model of Stem Cell Function

A theoretical model of stem cell function is presented here to further consider generalist agent function in a biological system. Stem cells function to maintain the integrity of the complete concept of the organism that is coded for in the DNA of that organism. Each individual agent or cell contains the DNA or source code of that organism, and the full expression of the DNA defines that organism's phase space, which can be referred to as an "attractor" or as that organism's "source code concept". Stem cells can migrate throughout the organism and interact with all the cells of the organism, either directly or indirectly. To interact with another cell, the stem cell must "badge" with the other cell by exhibiting appropriate surface proteins that are congruous with surface proteins on the other cell. After successful badging, the stem cell can functionally participate in the activities of the other cell, and can share information with the other cell. Stem cells have a relatively open source code, but specialized cells have much of their source code blocked and inaccessible. Because of a relatively open source code, stem cells can function with all specialty cells. As a stem cell functions with and as a specialty cell, it can collect information, some of which can be passed to the stem cell from the specialty cell. Also, because some information only exists dynamically, some information can be obtained directly by the stem cell through functional participation. The information that is collected by the stem cell is then evaluated by comparison of the information against the stem cell's relatively open source code for analogous correlations, and in this manner it can be determined if new information is novel, and if this new information is appropriate to the source code concept. The information can be evaluated as a whole and in parts, and as it is compared to the source code, binary decisions are made regarding the appropriateness at intervals. Each interval represents a gene or part of a gene, and an active decision is made at each such interval to determine whether or not there is a correlation at that interval. Such binary or "spin" decisions are made many times for a particular piece of information, and through this process the stem cell determines the orientation of the information in relation to the source code concept. If the information is deemed possibly useful by the stem cell, then the stem cell performs a functional trial to evaluate the usefulness of that information. If the information is deemed useful after functional trial, it is then archived by the stem cell. Archiving of useful information can take place in the cytoplasm in RNA, and archiving can also take place in nuclear DNA. (The major histocompatibility complex (MHC) is an example of an area in which archiving of successive useful solutions takes place, with each new solution improving upon the last.) Archived information can be passed on to other cells, both specialized cells and other stem cells. Some of the archived information is geometric reconfiguration that serves to block or unblock certain portions of the DNA source code. Archived information can be passed to germinal tissue through stem cell migrations and badgings. Stem cells function to effect the geometric reconfiguration of the source code in other cells, and thus control the blocking or unblocking of source code in specialty cells or in other stem cells. In order to function as a specialty cell a stem cell must have a portion of its source code blocked, but because stem cells function in the geometric reconfiguration of the source code, after functioning as a specialty cell that stem cell can interact with another stem cell with more open source code and have that portion of source code unblocked. Functional information that is passed to other stem cells can thus be tested by other stem cells.

Generalist Agent Algorithm

An algorithm has been developed that describes the process of generalist function in any multi-agent complex adaptive system. ("A Method for Governing the Operation of a Generalist Agent within a Multi-Agent Complex Adaptive System." U.S Patent and Trademark Office application number 12/046,781; filing date: March 12, 2008.) This Generalist Agent Algorithm functions to maintain the integrity of an optimal source code concept within the function of the various agents that are chartered by the source code, while organizing information generated by those agents as to that information's usefulness and novelty. An agent utilizing this algorithm functions by migrating among specialty agents and, after appropriate badging verifications, participates in the functions of the specialty agents while collecting the shared information unique to individual subsets of specialty agents. This information is then evaluated against the source code for fit and/or novelty, and if found appropriate leads to a reconfiguration of the source code. This reconfiguration is then functionally tested, and if found useful is archived by the generalist agent, and then shared with other agents (both specialty agents and generalist agents), as the generalist agent migrates among them. Generalist agent function occurs within multi-agent complex adaptive systems, and within multi-agent complex adaptive systems of systems, in which the various agents that make up such systems possess a multi-dimensional source code, and in which those various agents deal with information outside of that source code. The generalist agent algorithm functions to generate, distribute and effect useful solutions within the system that are in accordance with the overall goals of maintaining the integrity of the optimal source code concept.

The Generalist Agent Algorithm (GAA) is a function of an agent that exists within a multi-agent complex adaptive system. The characteristics of the multi-agent complex adaptive system are that it is made up of individual agents that are arranged multi-dimensionally, and that all possess the same source code, and that source code can be reconfigured geometrically or amended in any or all of the agents. Source code is a multidimensional array of information that includes executable functions and/or instructions (e.g. computer programs, nucleic acids, articles of incorporation or constitution, etc.). Agents in the system may differ according to what portions of the source code are open for access by the agent, and what portions of the source code are blocked from access by the agent. The blocking of source code is a reversible process, as is unblocking of the source code. The source code can be accessed or read linearly, but is also multidimensional and as such can be reconfigured in multiple dimensions while retaining its quality of linear accessibility, and such reconfigurations can provide different linear readings. Source code blocking and unblocking, and source code amendment, are functions of generalist agents, and take place through the Generalist Agent Algorithm.

Generalist agent and specialty agent are relative terms that denote a relative accessibility of the source code of the two agents. While both agents would have the same total source code, the generalist agent would have much more of it accessible and less of it blocked than would the specialty agent.

The Generalist Agent Algorithm can be represented in the following flow diagram:

```
→→→→→→ migration
↑               ↓
↑            badging
↑               ↓
↑            functional participation←←←←←←←←←←←
↑               ↓                               ↑
↑            information collection             ↑
↑               ↓                               ↑
↑            information evaluation             ↑
↑               ↓                               ↑
↑            functional trial→→→→→→→→→→→→→→→
```

```
↑                    ↓
↑               usefulness decided
↑                    ↓
↑               archiving
↑                    ↓
←←←←←←  distribution
```

In the migration phase of the algorithm the generalist agents have the capability to migrate throughout the entire multidimensional system and come in contact with the other agents of the system. There is no hierarchy among generalist agents. In the badging phase of the algorithm the generalist agent exhibits evidence of source code configuration consistent with that of the particular specialty agent contacted. Because of a relatively open source code, the generalist agent can badge and function with all specialty agents. In the functional participation phase of the algorithm the generalist agent participates in the function of the specialty agent(s). The generalist agent, with its broadly open source code, has the ability to function as a specialty agent (although probably not as productively) and thereby can obtain information that only exists dynamically for those specialty agents. In the information collection phase of the algorithm the generalist agent collects such dynamic information from functional participation, and also collects information that is shared with it by like-badged specialty agents that are in close proximity. The information evaluation phase of the algorithm involves comparison of the information against the source code for analogous correlations, to determine relevancy and/or novelty of data as appropriate to the source code concept. The data is evaluated via the Information Evaluation Algorithm (discussed below) across the broadly open source code and multiple dimensions that are available to the generalist agent. The result of the Information Evaluation Algorithm is a new configuration of the source code of the generalist agent. In the functional trial phase of the Generalist Agent Algorithm the generalist agent functions as a specialty agent with the reconfigured source code to assess the new configuration for functional efficacy. In such function further information can be collected and evaluated, and other source code configurations can be assessed. In the usefulness decision phase of the algorithm, a new source code configuration that has been determined to be of efficacious function after assessment in functional trial(s), is found to be a useful and appropriate adaptation. An appropriate adaptation is a reconfiguration of source code that provides a linear reading of the source code that lowers entropy with respect to the information encountered. In the archiving phase of the algorithm, useful, appropriate, and significant adaptations of the source code are archived and made permanent. In the distribution phase of the algorithm there is a sharing of useful, appropriate, and significant source code adaptations with other agents, through contact with those other agents. Distribution may take place by the migration of the generalist agent to specialty agents, or may take place by initiation of a new line of specialty agents. Generalist agents can migrate to other generalist agents and share data. A portion of source code that is blocked in one generalist agent can thus be unblocked by another generalist agent that has that portion of source code unblocked.

The Information Evaluation Algorithm is a sub-algorithm of the Generalist Agent Algorithm that describes how data is evaluated with respect to the source code. The Information Evaluation Algorithm can be represented in the following flow diagram:

→→→→→→source code configuration codes for a recognition instrument specific
↑ to certain data
↑ ↓
↑ engagement of recognition instrument specific to certain data
↑ ↓
↑ specific data recognized in data stream by recognition instrument
↑ ↓
↑ activation of the portion of source code that codes the recognition instrument
↑ ↓
↑ linear signaling with other activated portions of source code (This linear signaling between activated portions of source code may occur across dimensions, and may occur between agents in relatively close proximity.)
↑ ↓
↑ coordination with other activated portion(s) of source code
↑ ↓
↑ multidimensional reconfiguration of source code
↑ ↓
←←←←←←blocking or unblocking of source code

In the Information Evaluation Algorithm the source code codes for a recognition instrument that is specific to certain data and then that recognition instrument is engaged to function in a data stream, and when the specific data is recognized by the recognition instrument, and this leads to activation of the portion of the source code that coded for the specific recognition instrument. Activated portions of the source code can linearly signal each other and coordinate a multidimensional reconfiguration of the source code that leads to a blocking or unblocking of the source code.

Specialty Agents vs. Generalist Agents

While all agents within a multi-agent complex adaptive system possess the same source code, they are differentiated from each other by the geometric configuration of that source code, which essentially blocks (or unblocks) specific portions of that source code. Such differentiation allows certain specialty agents to carry out certain functions with optimal efficiency. Such specialty agents have source code that is reconfigured for the optimal linear access, reading, and function of specific portions of the source code. Generalist agents, on the other hand, maintain an unblocked source code and thus can maintain the entire source code concept of the system within their scope of function. While a specialty agent has a limited scope of function that optimizes utilization of a specific linear portion of source code, a generalist agent maintains more multidimensional source code capability. Specialty agents are oriented toward specific linear source code function and so are necessarily resource-oriented, while generalist agents are oriented to the broader function of the system's overall source code concept and are more information-oriented. Generalist agent and specialty agent should be considered to be relative terms because agents in most complex adaptive systems utilize both specialty (or linear) source code functions and generalist (or multidimensional) source code functions. In biological systems it would probably be rare for an agent to be a purely specialty agent or a purely generalist agent, but rather an agent would possess both capacities while overwhelmingly expressing one or the other orientation. In most complex adaptive systems, a specialty agent possesses capability for both generalist functions and specialty functions, but is overwhelmingly oriented toward performing a specific specialty function (and thus is oriented toward obtaining and utilizing resources in this capacity). Likewise, a generalist agent possesses capability for, and engages in, both generalist functions and specialty functions, but because of a more broadly open source code it is more oriented toward the overall system goals (and thus is oriented toward information about the system and its conditions). An agent's capacity for generalist function is determined by its amount of unblocked source code, and blocked source code limits the expression of generalist function.

Adaptation vs. Compensation

In the face of changing operational conditions, modification of biological systems occurs with the result that those systems can continue to function despite the encountered perturbations, and different terminology has been associated with the description of such modifications such as "emergent", "dynamical", and "self-organizing". In complex adaptive systems this quality of adaptation differs from simple compensation in that while both adaptation and compensation are system responses to a perturbation, compensation is a response that is linearly coded for within an existing source code configuration, while adaptation indicates a reconfiguration of the system source code. The compensatory response of a biological system to a perturbation can be represented as a "resilience curve" which plots a measure of a perturbation against a measure of a related compensatory response.

The Frank-Starling Curve is a well-known example of such a resilience curve from cardiac physiology. It is a plot of the system perturbation of changing left ventricular filling pressure (left ventricular end diastolic pressure or LVEDP) against the system compensatory response of left ventricular stroke volume (SV). (Alternately the Frank-Starling Curve can be represented equivalently by plotting the right atrial pressure against the cardiac output.) The Frank-Starling Curve shows how the agents (in this case heart muscle cells or myocytes) that make up the left ventricle of the heart work together to contribute to the maintenance of circulatory function in the face of the fluctuating environment of varying left ventricular filling pressures. An important factor in understanding the Frank-Starling Curve is that the heart muscle cells contract more forcefully the more that they are stretched prior to the moment of contraction, but that there is an obvious limit to the stretch distance that can be accommodated by this mechanism, and this reactive mechanism begins to lessen and then fail as the heart muscle cells become over-stretched. The plot of this on the Frank-Starling Curve ascends in a positive-sloping fashion in the section of the curve in which the full compensatory function of the heart muscle cells is active, but the curve reaches its peak as the limit of the compensatory stretch function is reached, and then as the heart muscle cells become overstretched, the slope of the Frank-Starling Curve rapidly turns negative indicating system decompensation and leading to rapidly decreasing left ventricular output in the face of increasing left ventricular filling pressure.

```
                     |
                     |              ***
                     |            *     *
compensatory         |          *        *
response             |        *          *
(SV)                 |      *            *
                     |    *              *
                     |_*_________________*____
                          perturbation
                           (LVEDP)
```

The Frank-Starling Curve can be considered as a prototypical resilience curve because it shows how a multi-agent system compensates for an environmental fluctuation (i.e. system perturbation) and how its compensatory mechanism can eventually be overcome by severe fluctuation. Such a curve is a characteristic of biologically-based multi-agent systems, and illustrates the maintenance of a compensatory or resilient steady state within such a system. While such a plot represents compensation of a system, it does not represent adaptation of that system.

It is to be remembered that *in vivo* the Frank-Starling Curve is actually a series of curves that vary with different cardiovascular states such as afterload (i.e. blood pressure) or inotropic state (i.e. strength of contractility of cardiac muscle). Each different inotropic state or afterload defines a different shifting of the Frank-Starling Curve for that system. The heart itself can undergo a remodeling (i.e. structural change) over time that will reconfigure cardiac geometry to allow more efficient and effective function to deal with long-term changes in inotropic state or afterload, and such remodeling also shifts the Frank-Starling Curve for the system.

While an individual Frank-Starling Curve characterizes compensatory function for the system, it is the shifting of that curve that characterizes adaptive function. Adaptive function thus defined implies a multidimensional reconfiguration of the system that would be correlated with a multidimensional reconfiguration of the source code. This allows differentiation between simple compensatory functions and more complex reconfigurations of source code that result in true adaptations, and the reconfiguration of source code is dependent upon the function of generalist agents.

Flocking

Flocking systems have been frequently used as examples of multi-agent complex systems, and as such can be used to illustrate application of resilience curve functions in the definition of adaptation. Consider a one-dimensional flocking system (e.g. cars in a line of traffic) in which the individual agents are programmed through their source code to maintain a tail-to-nose following distance (d_f). To maintain this following distance the agents must speed up (i.e. accelerate) when the following distance is greater than d_f and slow down (i.e. decelerate) when the following distance is less than d_f. A resilience curve for this compensatory system would be a plot of the system perturbation of a changing following distance against the system compensatory function of acceleration. Now consider that an unanticipated higher dimensional perturbation is presented to the system in that a projection of length d_p extends from the tail of each agent, and that despite the projection, the agents still determine following distance from nose to tail without recognizing the added distance of the length of the projection, and that the added distance of the projection causes collisions between agents which inhibits successful flocking. It should be noted that d_p exceeds the natural variability in following distance for the system. The agents need to correct the following distance to reflect the greater following distance that should now be ($d_f + d_p$). The source code contains no linear coding for correcting the greater following distance necessary for successful flocking. A reconfiguration of the source code is necessary to adjust the following distance, and so somewhere in the source code there is a reconfiguration that leads to a modification of a Euclidian distance in the source code (d_c) that is expressed phenotypically as a lengthening of the following distance, and this is reflected as a shifting of the resilience curve for the system. So simple flocking by itself is "compensatory" but it in itself is not "adaptive". Geometric reconfiguration of the source code is necessary to effect adaptation reflected by a shift in the resilience curve.

Mechanisms exist for modification of the system's source code (possible mechanisms might include geometric reconfiguration, amendment, or random mutation), and it can be postulated that the lengthening of the following distance could be coded for geometrically within the source code rather than linearly. For example, a linear source code reconfiguration could occur that produces a two dimensional loop in the source code that might lead to a new linear reading of the source code. The size of the loop generated in such a geometric reconfiguration could be correlated with the following distance. In such a case there is necessarily a relevant Euclidean distance in the source code that is correlated with the following distance, and a change in that relevant Euclidean distance of the source code would correlate with a change in the following distance (e.g. [$\Delta d_c / \Delta d_f$]).

It is known that eukaryotic DNA contains repetitive sequences that do not code for any protein, and these repetitive sequences have been referred to as "junk DNA". Such repetitive sequences may represent "ratchet points" of geometric DNA reconfiguration that have corresponding geometric phenotypical expressions. Considered in this manner, "adaptation" occurs by a multi-dimensional reconfiguration of source code, and it can be expressed in terms of a shifting of a "resilience curve" that resembles the Frank-Starling Curve.

Building Order

A characteristic of biological systems is that they build order as defined by the source code of their DNA. Biological systems seem "counter-entropic" in that they build order in the midst of a universe that trends toward increased entropy, but they do not violate the Second Law of Thermodynamics because they operate in an open system, whereas the Second Law of Thermodynamics is predicated upon a closed system. In the open system in which biological systems function, energy is input to build order.

If pressure is applied to a crystal of calcium, then over time that crystal will tend to break down. However, if pressure is applied to a crystal of calcium within the biological system of a bone within a living organism, then over time then that crystalline structure of bone will actually strengthen. (It is interesting to note that if too much pressure is applied too quickly, then the bone will break, and this will produce a resilience curve [pressure vs. bone strength] with a sharp negative slope similar to the failure of the left ventricle in the Frank-Starling Curve.)

Pressure on living bone induces the compensatory response of a change in the calcium crystal lattice structure. This change in the geometrical configuration of bone is not coded for directly or linearly in the source code, but rather occurs as a result of a geometric reconfiguration of the source code that is expressed phenotypically in the new geometry of the bone. As the geometry of the bone changes, the resilience curve for the bone shifts to reflect the new characteristics of the remodeled bone. It is by generalist function that the generalist agents of a system (e.g. stem cells) integrate and coordinate adaptive source code reconfiguration.

Entropy

In a biological (adaptive) system energy is input to build order. Entropy is a measure of disorder or uncertainty. According to Shannon information theory, entropy is a measure of information content that the recipient is missing when the recipient doesn't know what comes next. In a multi-agent system that is organized in a pyramidal fashion (e.g. a corporate organization chart or a Phylips tree diagram), as the Euclidian distance between two points (and/or the number of nodes) increases arithmetically, the entropy of the system increases logarithmically. There are (binary) decisions to be made at each node or level of the system, and the uncertainty surrounding these decisions is frequently managed stochastically via statistics, vector summation, or heuristics, however, in such systems as the Euclidean distance or number of nodes involved increases, the entropy involved quickly approaches the point of total random disorder. Such disorder would be incompatible with the continued existence of the biological system, but biological systems overcome such potential increases in entropy through the generalist function of generalist agents utilizing the Generalist Agent Algorithm. In such pyramidal systems generalist agents function to lower the system entropy and allow appropriate processes to proceed, in an analogous way to a catalyst lowering the entropy of a certain chemical reaction. It is also analogous to the function of an enzyme in a biochemical system that lowers the entropy for a specific pathway that allows a specific biochemical reaction to take place in the midst of a milieu of multiple potential biochemical pathways. Overcoming entropy is what lowers the necessary energy of initiation of a specific pathway in a chemical system, or in a biochemical system, or in a multi-agent complex adaptive system. It is the generalist agent that acts to enable a type of "selective catalysis" that allows a specific process to proceed within a multi-agent complex adaptive system, by lowering the entropy specific to that process through a reconfiguration of the source code via the Generalist Agent Algorithm.

An illustration of such "selective catalysis" by generalist agents can be envisioned in a corporate pyramidal structure described by an organization chart. In his book entitled Warning Analysis for the Information Age: Rethinking the Intelligence Process the author John W. Bodnar points out (on page 88) how the increased "degree of assembly" of a system's organizational structure leads to increased entropy that raises the "activation energy" of decision making within the organization, and thus inhibiting a decision being made by the system. It is often difficult for appropriate information to move from the lower levels of a large organization to the upper levels where decisions are made, and it is likewise often difficult for information to move laterally in such corporate structures. A generalist agent can migrate throughout all aspects of the corporate structure and coordinate information exchange while integrating that multisource information with the overall goals of the corporation (i.e. the source code concept of the system). Overcoming entropy by coordinating information flow lowers the necessary decision making "activation energy" for processes appropriate to the source code concept of the system, and thus in effect selectively catalyzes appropriate decision making. Generalist agent function allows the linear system functions to proceed within the typical corporate structure, while providing for those functions to be superseded by selective catalysis of pathways that work around the linear functions by selectively lowering system entropy. This provides for adaptation of the system with respect to the source code concept.

Applications

Generalist agent function and the utilization of the Generalist Agent Algorithm is a necessary component of any multi-agent complex adaptive system, and it has application in computer systems, organization and management of human organizations, and bioscience.

There is a growing array of computer systems that are based upon intelligent agents. Intelligent agent systems are vital for computer systems involved with information management and knowledge discovery (IM&KD). Generalist agent function within such programs can enable improved discovery of relevant data from massive sources because such activity is the essence of generalist agent function. Data visualization programs can be enhanced by generalist function-guided multi-dimensional link and cluster analysis. IM&KD warning analysis could be improved through identification and tracking of malevolent generalist agent entities.

Other computer system applications of the Generalist Agent Algorithm center around the organization of large computer router networks. The developing explosive growth of mesh networks, mobile routers, and "cloud computing" require new paradigms of organization for large systems such as the envisioned "smart electrical grid". The Generalist Agent Algorithm also presents significant implications for computer research in "neural networks", "biologically inspired cognitive architectures", and "artificial intelligence". The multi-dimensional analytical capabilities of the Generalist Agent Algorithm have applicability to new multi-dimensional generations of computer technology, such as parallel processor chips, 3-D memory chips, and memristors.

The Generalist Agent Algorithm also has great potential for application in the field of organization and management. With the paradigm of generalist agents and generalist function in mind, more resilient complex structures can be developed for military units dealing with the considerations of fifth generation warfare and the networking of forces. With a conscious effort toward enabling generalist agent function, the Intelligence Community can organize to prevent the filtering out of actionable intelligence through improved analysis of data at lower organizational levels. The Generalist Agent Algorithm dovetails with many aspects of the concepts of "analytical pathologies" and "alternative analysis" that have been proposed within the Intelligence Community. Generalist agent theory naturally extends to the realm of economics, and especially to market analysis and prediction.

The engineering of so-called "distributed systems" has been hampered by the lack of good modeling of such systems, but the engineering of such systems will need to take place beyond just hardware and software, and move toward integrating design, implementation, and education. Generalist agent theory is applicable here, as well as in social network research (the study of purposeful collectives), mathematical biology research (the study of mathematical models of adaptation and response), evolutionary computation research (the study of decentralized problem solving), and systems engineering, especially when system resilience considerations are important.

The bioscience applications of Generalist Agent Theory are obvious, and extend at least to genetics and ecology, and probably to oncology and beyond. Applications within genetics to the fields of immunology and transplant would be obvious, but ecological theory dealing with systems interactions would also be advanced. Analysis of system pathologies will demonstrate dysfunctions associated with deficiencies in generalist function, and specifically the potential for the malignant transformation of a multi-agent complex adaptive system that undergoes modification of its source code without the coordination of adequate generalist function.