A SYSTEMS ENGINEERING APPROACH TO MODELING AND SIMULATING SOFTWARE TRAINING MANAGEMENT EFFORTS

Asli Soyler Akbas[(a,b)], Waldemar Karwowski[(b,c)]

[(a)] Institute for Simulation and Training
[(b)] Institute for Advanced System Engineering
[(c)] Department of Industrial Engineering and Management Systems
University of Central Florida, Orlando, FL, USA

[(a,b)]asli.soyler@ucf.edu, [(b,c)]wkar@ucf.edu

ABSTRACT
Although being directly affected by the fluctuations in complex adaptive systems such as knowledge transfer, and economy, technology training within organizations, are managed as independent projects, furthermore causing critical information, as requirements and scope changes, being failed to be shared. This existing approach, forces only current status to be used with discrete data in decision making rather than evaluating the continuous behavior of the training process integrated with possible future environmental conditions. The purpose of this paper is to initiate the design of a model for understanding the behavior of complex technology training management system (TTMS). Recognizing the process as adaptive and continuous, this paper captures the ongoing efforts to simulating training management efforts that can support organizations in critical decision making, and requirements and risk management using system dynamics and agent-based simulation designed with model-based system engineering approach.

Keywords: Training management, SysML, system dynamics, agent-based simulation

1. INTRODUCTION

Today, software solutions are used aiming to support employees with almost every task, related to their area of work. Due to vastly improving technology, stakeholders are often forced to improve the existing software packages in order to satisfy the arising needs and improve the work process. While some of the improvements stay to be as existing version upgrades, in cases where new software is selected, depending on the project scope, the training process may involve the majority of employees within that organization.

Defined as complex adaptive social systems (Morel and Ramanujam, 1999), organizations behave in motion of rotating circles, building a continuously repeating curve with three high level states as Equilibrium, Dissolution and Growth, which can be seen in Figure 1 (Marten, 2001). Besides the external triggers affecting the state of equilibrium, by changing the requirements of the tasks and their approach, adapting a new software solution that affects the majority of the organization will create an internal trigger causing the state to change to dissolution. At this state, effectiveness of transformation efforts will define the duration until the Growth state is achieved, or in worst cases, drive the system in to chaos.

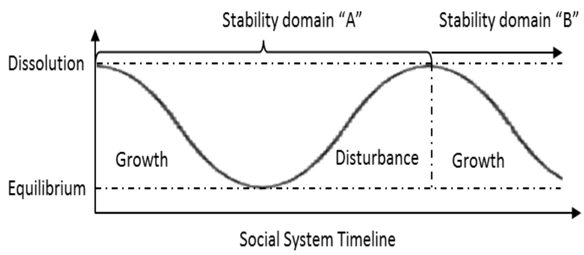

Figure 1 Social System State Adapted from Marten, G. G. (2001)

The importance of training efforts through organizational transformation has been emphasized in literature (Kezar and Ecke, 2002). However, the current body of knowledge lacks research on studies modeling and simulating technology training management system (TTMS) as complex adaptive even though the knowledge transfer within training is recognized as one (Burns and Knox, 2011)

2. BACKGROUND

The key predictors of training transfer are grouped in three as (1) immediate training climate (Kontoghiorghes, 2001), (2) Trainee personality traits and characteristics (Colquitt et al., 2000), and (3) overall organizational environment (Baldwin and Ford, 1988) such as teaching methodology, self efficacy and immediate peer support, respectively. Furthermore effect of (4) economy (Bass and Voughan, 1966, Galf and Hammour, 1993), and (5) technology improvements (Helpman and Rangel, 1999) on training knowledge transfer were noted in literature. A review of literature was conducted to further understand the individual behavior of these factors as systems interacting with training performance and their level of complexity. The review efforts were grouped in four areas as knowledge transfer-which included the two groups of training transfer key predictor factors (1 and 2)-organization, financial system, and technology dynamics.

The reviews in knowledge transfer dynamics supported existence of complexity in learning systems, a phrase introduced by Davis and Simmt (2003) describing collective classroom components, is advocated by also other researchers (Burns and Knox, 2011, Davis and Sumara, 2006). Additionally, Newell (2008) evaluated the potential benefits and challenges of accepting this theory following Davis, Simmt and Sumara's published arguments on how individual learner and teacher dynamics interacts and emerges as learning. On the other hand, organizations are accepted as "dynamic systems of adaptation and evolution that contain multiple parts which interact with one another and the environment" (Morel and Ramanujam, 1999). Furthermore, their nested structure continuously interacts with other macro and micro, systems and sub-systems, respectively (Folke & Folke, 1992). New systems may arise from emerging dynamics as part of the system, due to change processes occurring with an organization (Dooley and Van de Ven, 1999). Similarly, the large-amplitude and aperiodic fluctuations in economic variables were shown as evidence for its complex adaptive nature (Chian, 2007). Finally, technology management in organizations recognized as complex adaptive systems as it interacted with emerging and non-linear trends (McCarthy, 2003).

The characteristics of technology training management that matches the most common properties of complex adaptive systems (Bot, 2012) are as follows:

- Training management interacts with organization system and knowledge transfer variables (micro).
- Although there are techniques to support training planning often times changes in duration, cost, training performance occur.
- Training management is part of knowledge transfer system.
- Employees within an organization create a unique knowledge share structure creating a culture which emerges individual and organization's learning state (Weick , 1979)
- Training is applied in organizations in iterations, the lessons learned from each experience (outputs) feeds the following management strategy as inputs. (Armstrong, 2003)
- If started without well planning the effects of each variable and their interactions, training efforts will fail
- In training management, change in one variable, for instance organization's climate or available resources, will trigger a change in the whole system and will affect outcomes.
- Training management rely on the resource availability, depletion of any resource will trigger system's state to change to 'steady-state'.
- Training has emerged from interaction of systems such as learning, organization and technology. Through time its internal interactions derived management variables (Dooley and Van de Ven, 1999)
- Due to continuous change in it is variables such as humans and technology same management approaches will result in varying outputs.

These characteristics support our argument that instead of managing technology training using linear optimization techniques, complex adaptive systems theory processes should rather be used to understand and capture its patterns and interacting mechanisms.

3. METHODOLOGY

Today, the literature provides researchers with a great source of knowledge on studies related to technology training. Although these studies improved the understanding of the training systems, the vastly improving technology continuously been introducing new areas to the existing gap that's been already identified by the researchers (Salas and Cannon-Bowers, 2001). Furthermore, technology training management is yet to be recognized as complex adaptive in literature.

This research aims to develop a hybrid simulation model using mode-based system engineering (MBSE) approach which some of its benefits are captured as (Sage, 2009):

- Develops a unified coherent model
- Enables the realization of successful systems
- Defines needs
- Documents requirements and
- Facilitates the interoperability between people and organizations (Ramos et. al, 2012)

This approach will support efforts in representing technology training behavior and management processes, driven by the requirements arising from its interactions with other systems knowledge transfer, organization, financial, and technology. The research efforts will be grouped in four main areas.

3.1. SysML Model Development

The overall design will propose a unified coherent model (Sage, 2009) that will define and capture technology training management system by studying its structure, and behavior shaped from its stakeholder and system requirements. Systems Modeling Language (SysML), which facilitates systems engineering activities that are concerned with the whole, complexity, multi-disciplinary and holistic thinking and synthesis (Ramos et. al, 2012) and its four pillars will be used in modeling the system that are captured in Figure 2 (Hause, 2006). Traceability, will be one of the key benefits of this approach that later will allow establishing the corresponding feedback loops within the system dynamics model.

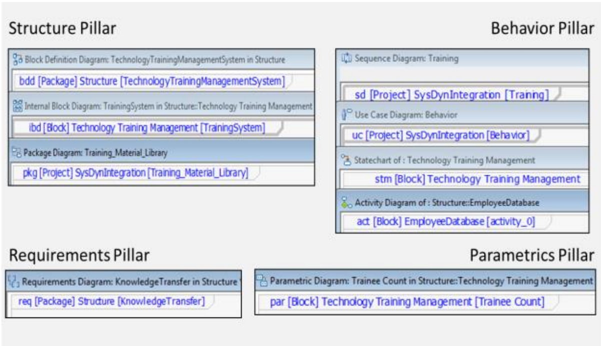

Figure 2 Four Pillars of SysML Adapted from Hause, M. (2006)

3.1.1. Structure Pillar

Structure pillar diagrams-which may be considered as the building block of the overall system, as they capture and map the components and boundaries of the system - including sub-systems, their components and the possible interactions among each using Block Definition Diagram (BDD) and Internal Block Diagram (IBD), respectively, will be designed in this phase. The overall system will be composed of three subsystems that are Project Management Office (PMO), Knowledge Transfer (KT) and Software as captured in Figure 3. This structure will allow capturing factors that directly and indirectly affect the training outcomes, identified in literature while maintaining a representation of the current structure of the organization. For example, the organizational dynamics and their effects, such as peer influence behavior, will be studied under PMO sub-system.

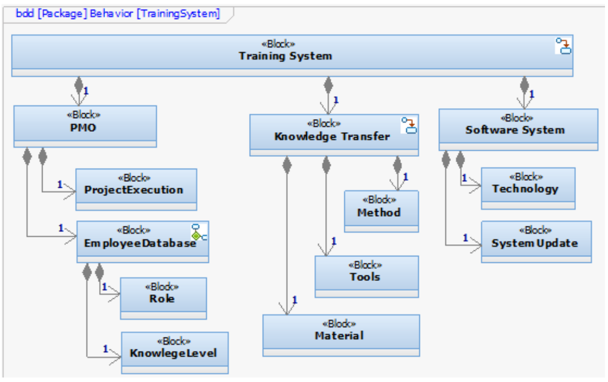

Figure 3 Block Definition Diagram (BDD)

3.1.2. Behavior Pillar

The second pillar, Behavior, consists of the diagrams that define how the model would behave within the capabilities of the designed structure. The design process in this phase will start by defining the states of the overall system, its subs-systems and their components using State-Machine Diagrams (STM). At the highest level, the TTMS is designed to have two states, serving as an "on/off" decision switch for the overall system. The states of the models are designed to be triggered by the factors such as "Identified trainee count", "Project budget" and "Project schedule" as seen in Figure 4.

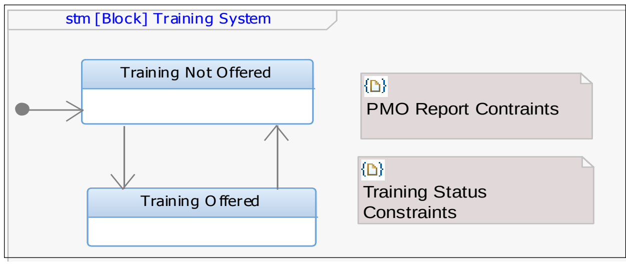

Figure 4 Highest-Level State Diagram (STM)

Later, starting from the highest level, the Use-Case (UC) diagrams will be built capturing the actions and the responsible actors, as groups of different stakeholder classes. Modeling of actors will be one of the most important steps in this section. Properties for each actor class will be filtered out according to criteria: employee type (trainer, trainee, project manager and other stakeholders), skill level (novice, advanced beginner, competent, proficient, and expert), employee position (manager, engineer), department, training attendance count and training transformation factors (demographics, personality, cognitive capabilities and so on). The Package diagram will be used to capture the class hierarchy and the attributes of each agent sub-classes. Furthermore, individual UC diagrams will be linked according the assigned "composition" relations creating traceability among all diagrams. The following step will include capturing the functions included within use-cases, occurring within and in between these components, which will be designed using the Activity Diagrams (ACT). The use of Sequence Diagrams will be decided after completion of the ACT.

3.1.3. Requirements Pillar

The Requirements pillar will allow establishing the rules of the system, rather than using long descriptive paragraphs, the requirements will be captured as short, testable statements. Later, the traceability between child and parent requirements will be captured using "Refine/Refined by" relation. Furthermore, each will be linked as conditions to states of TTMS components which will define the behavior of the system.

3.1.4. Parametrics Pillar

The parametric diagrams allow capturing and modeling constraint expressions, representing system constrains derived from the requirements. As the initial step, the structure, hierarchal relations, of the constraints, which will be represented using type-specific block, will be built using the BDD diagram from the first pillar. After establishing the parent-child relationships using "composition" type, the parameters will be assigned according to the constraint logic statement. Similar to the process followed in IBD, each constraint block will be connected using the input/output ports within the Parametric Diagram (PAR).

3.2. System Dynamics Model Development

The principles of system dynamics modeling, such as the ability to study the effects of individual variables and their interactions, provide a pragmatic and holistic nature (Romme and Dillen, 1997) that is found useful in modeling humans as social systems that are characterized by "dynamic complexity" (Senge, 1990).

The state charts for the system dynamics model will allow creating conditional rates between stock variables. For example, it will be possible to set the training rate to 0 if there are budget cuts, or similarly, have the ability to arrange it according to the training demand. The states of the components created in the SysML model will identically represent the states in system dynamics model (Figure 5).

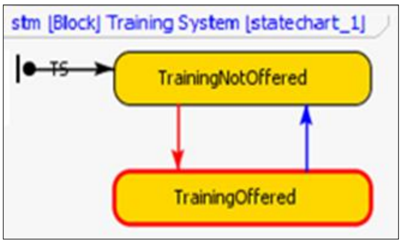

Figure 5 Highest-Level System Dynamics States

Similarly, the states of components, which have assigned constraints, such as skill acquisition and attendance will be created as stock variables and corresponding state change conditions will be used as conditions for the flow rates within the system dynamics model as shown in Figure 6.

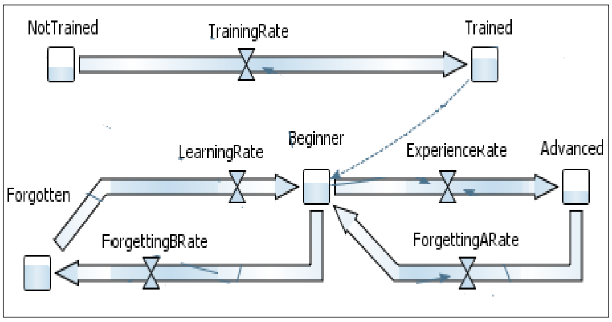

Figure 6 High Level Stock and Flow Variables

Furthermore, properties assigned to actors such as demographics, department and so on within SysML will be the variables of the system dynamics model. Their causality coefficients will be assigned according to the meta-analysis and experiment results collected from literature. Figure captures an example of an actor and the corresponding dynamics of the variable "ComputerSkill" modeled with AnyLogic which was adapted from the experiment conducted by Harrison and Rainer (1992). Additionally, the coefficients of the significant factors found in the regression analysis, indicated by rectangles, were used as factors that calculate the ComputerSkill variable. It is important to note that, rather than correlation, significant predictive factor coefficients collected from regression analyses, will be used to as weights of the variables in the system dynamics model. The variables in system dynamics are connected to one another by a cause-effect type relation thus the correlations collected from a study cannot be used as a coefficient of a variable.

3.3. Agent-Based Model Development

The limitation of system dynamics modeling, which is the missing capability to capture the properties of observed entities and the resulting effect of these differences (Bonabeau, 2002) will be supported using the agent-based model of TTMS. The hierarchy of the agents will mirror the actors previously created in the SysML model. Although stakeholders, such as software developers and project managers are involved in the overall system, they will not be created as agents. The primary reason is that their decisions may trigger a state change in simulation, but they don't directly interact with either trainee or trainer agents within the scope of this system. Secondly, even though their decisions as inputs are an interest factor in this simulation their behavior is not. Thus, the two agent classes that will be included in the system are Trainee and Trainer. Furthermore, the rules of the agents will be driven from the assigned properties of the actors in the SysML model. Each trainee agent will be assigned with five state groups as employee position (manager, engineer), department, skill level, attendance count and training transformation factors properties.

3.4. Integration of Two Simulation Models

In this phase, the classes of agents will be created and linked under the system dynamics model object. The different properties of agents will be connected with the corresponding stock variable. For example, the skill level property will be connected to the stock variables representing the acquired skill levels. Similarly, count of training attendance property will be linked to the "Not Trained", "Trained" stock variables. With this connection, additional to being able to capture the emerging behavior of the social group, it will be possible to differentiate count of employees from each department within a stock variable at any given time during simulation run.

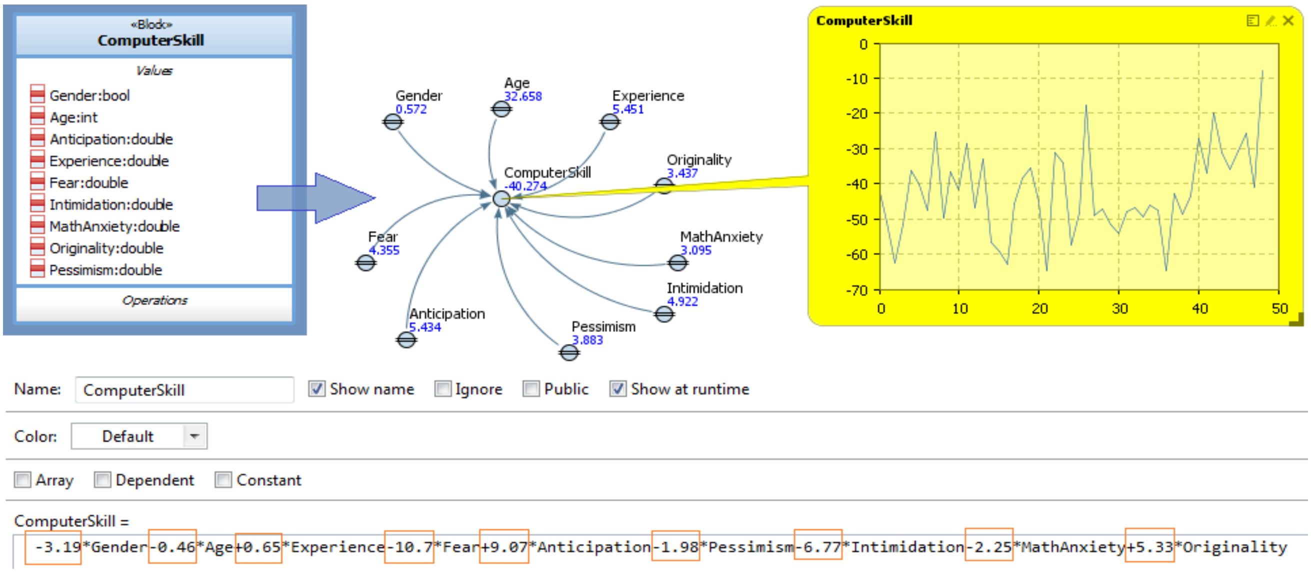

Figure 7 System Dynamics Design Example

4. TRAINING MANAGEMENT SYSTEM

The overall version of the proposed model will be a simulation platform, which will support the decision makers of an organization in planning and testing their training management strategies. The capability to expand upon existing capabilities of both simulation methods, of system dynamics and agent-based modeling, was the aim in selecting a hybrid model for development. The final simulation model will provide decision makers three variables, which will solely focus on high level project overview, cost, duration and the overall knowledge level of the identified trainees as can be seen in Figure 8. This output will serve as a quick evaluation of the decision alternative being tested. Additionally, descriptions of arising risks, unsatisfied requirements, and contributing factors will be provided for detailed analysis.

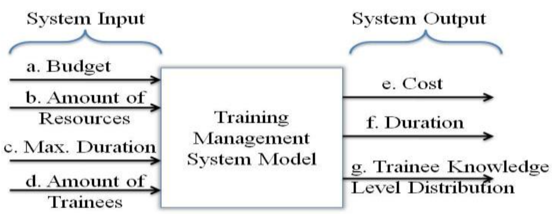

Figure 8 Black-Box Diagram of Overall System

4.1. Validation and Verification

The simulation platform efforts described in the Methodology section will be derived from the previously developed SysML model and depend on one another. As a result, each design and development phase will individually consist of system specific verification and validation processes, which will follow the methodologies captured by Kleijnen (1995).

The modeling will start from the highest-level (macro level) possible and add micro details in iterations. At each level the verification of the simulation models will include running simulations with deterministic values. Furthermore, different scenarios will be tested to check for any programming errors. The probabilistic values will be added once the behaviors of the models are verified.

At the end of phases, at first a validation test will be applied using the simple plot of the simulation output versus data collected from available literature. Immediate feedback on any inconstancies between the two samples will be evaluated. If the plot test does not show any variation to the naked eye, we will move to the hypothesis testing phase.

Each key predictive factor and their coefficients will be adapted from previous experiments published in literature. Thus, to validate a hypothesis test which will test existence of any significant difference between the correlations among two variables calculated from simulation and published correlations of each matching factor will be conducted.

5. CONCLUSION

The aim of this study was to initiate the design of a hybrid simulation framework that can support in understanding and managing technology training using model-based systems engineering approach. This research suggests that technology training management, rather than only being a process within technology transformation efforts, emerges as a component of an organization as it directly affects the outcomes and duration of achieving the state of growth while improving the organization's system stability. Furthermore, it is argued that studies using linear optimization techniques without feedback mechanisms are no longer usable due to the level of complexity involved in technology training management. Recognizing technology training management as a complex adaptive system, instead of managing efforts individually, a systems engineering approach to model its structure and behavior at an organizational level by studying its structure and behavior, driven by the requirements arising from knowledge transfer and its interactions with other complex systems was suggested. Additionally a methodology to establish a link between SysML and system dynamics and agent-based simulations was proposed.

REFERENCES

Armstrong, M. (2003). A Handbook of Management Techniques (p. 544). Sterling, VA.

Baldwin, T. T., & Ford, J. K. (1988). Transfer of training: A review and directions for future research. Personnel Psychology, 41, 63–106.

Bass, B. M., & Vaughan, J. A. (1966). Training in industry: The management of learning (p. 2). Wadsworth Publishing.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. Proceeding of the National Academy of Sciences of the United States of America, 99(10), 7280–7287.

Bot, K. D. (2012). Rethinking multilingual processing: From a static to dynamic approach. Third Language Acquisition in Adulthood (p. 82). Phillidelphia, PA. Retrieved from http://books.google.com/books?id=FDYls4wgX50C&printsec=frontcover#v=onepage&q&f=false

Burns, A., & Knox, J. S. (2011). Classrooms as complex adaptive systems: A relational model. TESL-EJ, 15(1), 1–25.

Chian, A. (2007). Complex Systems Approach to Economic Dynamics. Berlin: Springer Berlin Heidelberg.

Colquitt, J. A., LePine, J. A., & Noe, R. A. (2000). Toward an integrative theory of training motivation: A meta-analytic path analysis of 20 years of research. Journal of Applied Psychology, 85(5), 678–707. doi:10.1037//0021-9010.85.5.678

Davis, B., & Simmt, E. (2003). Understanding learning systems: Mathematics teaching and complexity science. Journal for Research in Mathematics Education, 34(2), 137–167.

Davis, B., & Sumara, D. (2006). Complexity and education: Inquiries into learning, teaching, and research. Mawah, NJ: Lawrence Erlbaum Associates.

Dooley, K. J., & Van De Ven, A. H. . (1999). Explaining Complex Organizational Dynamics. Organization Science, 10(3), 358–372.

Fleming, L., & Sorenson, O. (2001). Technology as a complex adaptive system: Evidence from patent data. Research Policy, 30(1), 1019–1039.

Folke, G., & Folke, C. (1992). Characteristics of nested living systems. Journal of Biological Systems, 1(3), 257–274.

Harrison, A. W., & Rainer, R. K. J. (1992). The influence of individual differences on skill in end-user computing. Journal of Management Information System, 9(1), 93–111.

Hause, M. (2006). The SysML Modelling Language. Proceedings of Fifteenth European Systems Engineering Conference.

Helpman, E., & Rangel, A. (1999). Adjusting to a new technology : Experience and training. Journal of Economic Growth, 4(1), 359–383.

Kezar, A., & Eckel, P. (2002). Examining the institutional transformation process: The importance of sensemaking, inter-related strategies and balance. Research in Higher Education, 43(4), 295–328.

Kleijnen, J. P. C. (1995). Verification and validation of simulation models. European Journal of Operational Research, 82(1), 145–162.

Kontoghiorghes, C. (2001). Factors affecting training effectiveness in the context of the introduction of new technology-A US case study. International Journal of Training and Development, 5(4), 248–260. doi:10.1111/1468-2419.00137

Levin, S. A. (2002). Complex Adaptive systems: Exploring the known, the unknown and the unknowable. American Mathematical Society, 40(1), 3–19.

Marten, G. G. (2008). Human ecology – Basic concepts for sustainable development (3rd ed.). Sterling, VA: TJ International.

McCarthy, I. P. (2003). Technology management a complex adaptive systems approach. International Journal of Technology Management, 25(8), 728. doi:10.1504/IJTM.2003.003134

Morel, B., & Ramanujam, R. (1999). through the looking glass of complexity: The dynamics of organizations as adaptive and evolving systems. Organization Science, 10(3), 278–293.

Newell, C. (2008). The class as a learning entity (complex adaptive system): An idea from complexity science and educational research. SFU Educational Review, 2(1), 5–17.

Ramos, A. L., Ferreira, J. V., & Barcelo, J. (2012). Model-based systems engineering : An emerging approach for modern systems. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 42(1), 101–111.

Romme, G., & Dillen, R. (1997). Mapping the landscape of organizational learning. European Management Journal, 15(1), 68–78. doi:10.1016/S0263-2373(96)00075-8

Sage, P. A., & Rouse, W. B. (2009). Handbook of Systems Engineering and Management (p. 847). Hoboken, NJ: John Wiley & Sons Inc.

Salas, E., & Cannon-Bowers, J. A. (2001). The science of training: A decade of progress. Annual Review of Psychology, 52, 471–499.

Senge, P. M. (1990). The fifth discipline. The art and practice of the learning organization. New York: Doubleday.

Weick, K. E. (1979). The social psychology of organizing (p. 32). New York, NY: Random House.