Automation System Design and Cognitive Systems Engineering

M. Kudret Yurtseven, Yeditepe University, Istanbul, Turkey
e-mail address: kyurtseven@yeditepe.edu.tr
Alireza Rahrooh, University of Central Florida, POBox 160000 Orlando,
FL 32816 United States, e-mail address: rahrooh@pegasus.cc.cf.edu
W. Walter Buchanan, Texas A&M University, College Station TX 77843
Texas, USA, e-mail address: buchanan@neo.tamu.edu
Seda Bas, Yeditepe University, Istanbul, Turkey
e-mail address: seda.bas@yeditepe.edu.tr

ABSTRACT

The aim in this paper is to look at some important issues involved in automation system design from the Cognitive Systems Engineering (CSE) perspective and propose a design framework. It is believed that the cognitive aspects involved in the design of automation systems are either overlooked or do not receive enough attention during the design process. It is suggested that the design be conducted within the framework of CSE, including human-machine relationships, which is vital for a successful design. The discussion developed in the paper explores the design issues within a proposed framework, based on GST (General Systems Theory) and CSE. In the framework, a soft *GST* system design approach is recommended as the "philosophical" guide for design; the Defense Acquisition Management life-cycle model is suggested for structuring the design process; Joint Cognitive Systems (JCS) paradigm of CSE is adopted for developing an appropriate control hierarchy. The paper emphasizes the theoretical aspects of the design process rather than the practical issues.

Key Words: Automation System Design, General Systems Theory, Cybernetics, Cognitive Systems Engineering, Defense Acquisition Management Life-Cyle Model, Joint Cognitive Systems Paradigm.

1. Introduction

Automation can be defined as the execution by a mechanism or a machine of a function that was previously carried out by a human (1). Its origins go back to water clock in ancient times; nowadays it is closely associated with computers and computing technology. The face of industrial automation changed significantly with the invention of ENIAC, the first digital computer in 1945. In the same year, Wiener established the foundations of *Cybernetics*, putting feedback theory and control, self-regulation and automation into a comprehensive perspective (2). Cybernetics is a strand of General System Theory (GST) where living systems are studied through analogy with physical systems. With the advances in technology, automation was expanded both horizontally and vertically in the 1960s. Large number of subsystems could be put under a common control, where each subsystem or process could be automated on several levels and layers; from minute process details to scheduling and production planning. Cost went down dramatically, hence automation could be applied everywhere, increasing human-technology interaction. This growing complexity inevitably changed the approach to automation design. In the beginnings, the design process was process–function centered; there was not sufficient concern for the human operator. The main theme was to introduce more automation to replace the "unreliable" human operator. As system control and automation moved from simple manual control to supervisory control, and then to fully automatic control, as time progressed, human-machine-relationship became ever more complex, forcing changes in the approaches to design.

In general terms, automation systems can be designed through one of the following approaches (3). In *The Left-over principle*, technological parts of the system are to do as much as feasible, from an efficiency point of view, while the rest are left to the operators to do. Here, the belief is that machines can do many things faster and more reliably, reducing labor costs. Humans are left with either simple jobs that are done rarely, or with very complex tasks that can not be done by machines. In this approach, the capabilities and limitations of human beings are not taken into consideration. In *The Compensatory Principle* or *Fitt's List*, capabilities and limitations of humans and machines are compared on a number of salient dimensions, such as speed, power output, consistency, information capacity (transmission), memory, reasoning/computation, sensing, and perceiving. Functions are then allocated to humans and machines in an "optimal" manner. *The Complemetarity Principle and Function Congruence,* on the other hand, became popular in the 1990s. It aims to sustain and strengthen the human ability to perform efficiently and therefore considers the work system in the long term, including how work routines change as a consequence of learning and familiarization. The main concern is not the temporary level of efficiency, but rather the ability of the system to sustain acceptable performance under a variety of conditions. This approach is consistent with *Socio-Technical System (STS)* design philosophy since it tends to look at the design issues as a human-machine cooperation rather than human-machine interaction.

The idea that human-machine systems or automation systems should be designed as socio-technical constructs were expressed by many researchers in the past. For instance, in 1977 Rousseau reported that "studies in job design and STS theory emphasize the importance of job characteristics to employee satisfaction and motivation, hence the success of the resulting system." (4). Martin, Kivinen, et.al. stated that "automation that is appropriate for application in realistically complex socio-technical domains should be based on an integrated understanding of the technical, human, organizational, economic and cultural attributes of the application" (5). The study reported in this paper falls into this domain; that is, automation and control systems are viewed as socio-technical constructs. As it is the case in modern system design, more than one approach or methodology is adopted; this is known as C*ritical Systems Thinking and Total Systems Intervention* (6), (7). All the approaches or methodologies chosen in this study are consistent with GST. A soft GST system design approach based on a series of ongoing *cybernetic* and fluid design functions is proposed as the general "philosophical" guide for design. For formulating and structuring the design process, *The Defense Acquisition Management life-cycle model* is suggested. It is also suggested that the relevant control hierarchy be based on the *Joint Cognitive Systems (JCS) paradigm* of CSE. The subject matter is presented in the following order in the paper. First, the important issues involved in the design of complex systems are explored within the context of GST. This is followed by a brief overview of the JCS paradigm. The last section includes the presentation of the proposed design framework, including the use of The Defense Acquisition Management life-cycle model and the JCS paradigm. The major findings of the study are summarized in the conclusions section. In the paper, the emphasis will be on the general and theoretical aspects of the subject matter rather the practical aspects.

2. General Systems Theory and System Design

Although the roots of General Systems Theory (GST) can be traced to Bertalanffy's work in microbiology in the 1930s, its formalization took place in 1954 with the foundation of International Society for General Systems Theory. The most prominent founders of the society were Ludwig von Bertalanffy and Kenneth Boulding (2). GST was developed, step by step, with the following goals: [a] to formulate generalized systems theories including theories of systems dynamics, goal-oriented behavior, historical development, and control processes; [b] to work out a methodological way of describing the functioning and behavior of systems objects; [c] to elaborate generalized models of systems (2). In GST, it was assumed that all kinds of systems (concrete, conceptual, abstract, natural and man-made) had common characteristics. However, GST is not a discipline – it is a theory cutting across most other disciplines linking closely. It uses various ways of classifying different types of systems, ranking in increasing order of complexity. Each level includes lower levels with its own emergent properties. In the various levels of taxonomy, it is the relationship between components in the system and not the nature of individual components that proliferate its properties and behavior. *Systems Science* is the applied form of GST, which is considered to be a *metadiscipline* with content capable of being transferred from discipline to discipline.

There are various trends in GST that can be adopted for analysis, control, and design of complex systems. All of these methods include a different view of system design. Systems science lead to development of *General Systems Thinking,* based on systems theory, helped generalists to handle complexity better than specialists. *Systems Approach*, on the other hand, is an application of systems theory to integrated framework of modern organizational knowledge and management science. It aims to treat all aspects of a human problem as a whole in a rational manner. *Systems Analysis* is another method which adopts a strictly systemic outlook on complex organizations. The aim here is to identify, reconstruct, optimize, and control an organization, while taking into account multiple objectives, constraints, and resources. Possible courses of action, together with risks, costs, and benefits are worked out, and the "best" course of action is chosen. *Systems Engineering* has become a very significant area of applications of systems science and systems thinking, particularly in the last decades. *INCOSE (International Council on Systems Engineering) defines* systems engineering as follows: Systems engineering is "an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem.*"* (8). Systems engineering has evolved since its foundation. Hitchins views systems engineering as follows: "it as a distinct discipline, founded on a system-scientific basis, including systems theory, systems thinking, systems engineering and systems management in a single framework" (9). He thinks that "systems engineering will expand its horizons to encompass all kinds and types of systems, including business, industry, socio-economic, governmental, and ecological." INCOSE proposes similar ideas in its Systems Engineering Vision 2020 document as follows: "The projected state of MBSE (authors' note: MBSE stands for model-based systems engineering) practice in 2020 will extend MBSE to modeling domains beyond engineering models to support complex predictive and effects-based modeling. This will include the integration of engineering models with scientific and phenomenology models, social, economic, and political models and human behavioral models." (10) - interested reader should consult the references (8) through (18) on systems engineering and MBSE.

The design problem taken up in this paper is closely related not only to systems engineering, but also to two other stands in GST: *Cognitive Systems Engineering (CSE*), and *Cybernetics*. CSE emerged as an interdisciplinary field, concerned with the study of work and design of human-machine systems in a socio-technical context. It requires a multidisciplinary approach, and is related to engineering, psychology, information sciences, management sciences, and computer sciences. Its components are cognitive tasks (thinking, problem solving and decision making), engineering, and systems. In CSE, the emphasis is on the overall system performance rather than on the functions involved. *Cybernetics* is concerned with studying living systems through analogy with physical systems. *Managerial Cybernetics* is also related to the theme of this paper, and it can be viewed as system science being applied as a problem solver in business organizations. The idea here is that modern managerial systems are based on an interchange between people, organizational entities and technical support. Therefore, it is impossible to separate social from technical factors; human being is always a part of the problem as well as a part of the solution. There are also some other system approaches, which are not a part of GST, but suitable for dealing with ill-structured systems. The descriptions and some applications of these approaches can be found in (6), (7), (19), (20), (21), (22), and (23). Jackson gives quite a comprehensive classification and critical reviews of these approaches from a systems management perspective in (6) and (7). In particular, Soft Systems Methodology, developed by Checkland, and Beer's Viable System Model may prove to be quite effective in designing the managerial subsystems of an automation/control system.

3. Joint cognitive system paradigm: a brief view

Comprehensive treatments of *Cognitive Systems Engineering* can be found in the books written by Hollnagel and Woods (3) and by Rasmussen, Pejtersen, et.al. (24). According to Hollnagel and Woods, CSE was formulated in the 1980s as a proposal to overcome the limitations of the information processing systems (IPS) paradigm in human-machine system design (3). They note that the IPS paradigm has been very popular in industry for a long time, in particular the S-O-R (Stimulus-Organism-Response) model. The Shannon-Weaver communication model is probably the most popular expression of the S-O-R model, and is known as the 'mother of all models'. This model is nothing but the black box model used in engineering where modeling is based on the observations of inputs and outputs. Here, a human is viewed as an IPS where mental processes are seen as finite state automaton. According to this view, the mental processes are considered rigorously specifiable procedures and mental states, which are defined by their causal relations with sensory input, motor behavior, and other mental states. It is claimed that this model is acceptable in information theory where one is interested in what happens to the signal as it is transmitted. It is also claimed that the IPS paradigm emphasizes *interaction*, and it produces a disintegrated view of the whole. The JCS paradigm, on the other hand, appears to focus *not on how parts communicate,* but on *how the joint system performs as a whole.* Humans and machine are seen as a joint cognitive system rather than as distinct but inter-connected components. Here, the human-machine relationship is viewed as *coagency* rather than interacting subsystems. One of the important consequences of this view is that system designers may have different 'models of the world', hence may develop different system designs. This very significant aspect of the JCS makes it a *pluralistic* rather than a monolithic paradigm. Pluralism is considered to be the major philosophical aspect that separates modern system approaches from the traditional ones (6), (7). In this scheme, machines and humans are not viewed as totally reactive and predictable where the unpredictability of machines may be due to machine dynamics or due to the operator's insufficient knowledge. For instance, an operator may loose control of a system if she/he does not follow the designer's suggestions, simply because the system was not designed according to the needs of the operator. In short, the idea is to design joint cognitive systems so that they can effectively control the situations where the functional work involved is cognitive.

The changes in the CSE paradigm are leading a fresher look at the ways of designing human-machine systems (3), (24). Firstly, a major shift in modeling philosophy seems to be occurring; a shift from Operations Research (OR) and System Science-based techniques to field study-based techniques. Secondly, optimization of system design is being based on the actual work content rather than on some misconceived hypothesis about the users. The reader should note that the use of OR and System Science–based techniques are not ruled out. Instead, some complementary CSE design tools are suggested to improve the quality of design. Thirdly, both vertical and horizontal communication lines should be established in the system, and the control structure should be decentralized. Fourthly, the system should have self-organizing features (where the system structure is updated as conditions change), or at least should have adaptive control features (where some of the system parameters are changed as the conditions change). Adaptation is essential since preplanned activities and procedures may not work well due to various reasons. Finally, like all human designed systems, the system must be goal-oriented, and the goals or objectives need to be propagated throughout the system effectively. Also, it is important to note that systems evolve over time, meaning that the objectives, the structure and the means of propagating these objectives have to be updated continuously. System capabilities and resources, in particular saturation of resources should be taken into account in the design process; system performance at the boundary conditions should receive careful consideration. This paradigm is applicable across a spectrum of single machine systems, socio-technical systems, and whole organizations, ranging from process and manufacturing industries, to military and service systems.

4. The System Design Framework Proposed

As mentioned earlier, it is suggested that several approaches/methodologies/models may be employed in the design process. It is thought that if these are used properly, the likelihood of producing a good design increases. In the design framework proposed, a soft approach is suggested to serve a "philosophical" guide to the overall design process. Within the framework, The Defense Acquisition Management Framework is suggested for mapping out the design process through a life-cycle model. Systems engineers will find this model rather friendly since they use these models in their professional lives. This particular model is chosen since it is comprehensive, systematic, and exhaustive (25). Also, within the framework, the ECOM (Extended Control Model) model of the JCS paradigm is suggested, which is to be used in the development of an appropriate automation/control system hierarchy.

4.1 The General Approach to System Design

The design of automation/control systems may involve human factors at a considerable level. Therefore, the design approach selected should have sufficient flexibility or softness to accommodate human aspects of the system. Among the methodologies that are oriented towards design of complex systems, the one developed by J. P. Van Ginch appears to be most appropriate for the purpose stated in this paper (2). This particular approach is soft in nature, hence can be adapted to various different situations. It is made up of a series of ongoing, cybernetic and fluid design functions. The approach involves ten steps, divided into three phases. These three phases are shown in Figure 1.

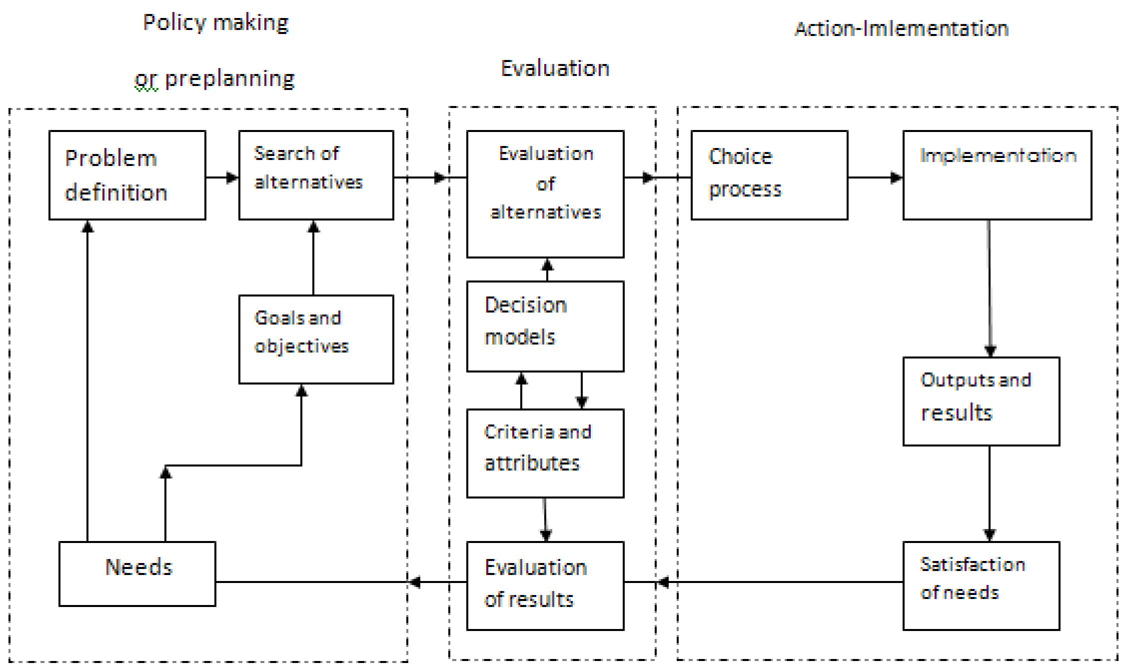

Figure1 The three phases of systems design

The following is a brief description of the phases and the steps involved:

First Phase: Policy Making and Planning
Step 1: Problem Definition:
A. The recipients or clients whose needs are to be met; B. The needs to be met; C. The scope, the extent to which needs to be satisfied; D. The agents involved: all those influenced by the project, considering their interests; E. Evaluation of the agent's world-views according to Step 2; F. The methods: short and general descriptions of methods which will be used to solve the problem; G. The system's boundaries: these should be defined, together with assumptions or constraints affecting the solution or its implementation; H. An enumeration of available resources compared to resources needed; I. A disclaimer to restrict hopes that systems design will provide a solution to everyone's problem.
Step 2: Understanding the world-views of clients and planners:
A. Premises; B. Assumptions; C. Values; D. Cognitive style.
Step 3: Goal Setting:
A. Needs and wants; B. Expectations and aspiration levels; C. Substitutions, tradeoffs and priorities; D. The morality of systems design (ethical issues).
Step 4: Search for and generation of alternatives:
A. Program alternatives and agents relationships; B. Determination of outcomes; C. Consensus.

Second Phase: Evaluation
Step 5: Identification of outputs, attributes, criteria, measurement scales and models
A. Identification of outputs; B. Identification of attributes and criteria; C. Determination of measurement scales; D. Choice of measurement models; E. Determination of the availability of data.
Step 6: Evaluation of alternatives
A. Use of models; B. Measurement of the outputs of soft systems.
Step 7: Choice of alternative

Third Phase: Action Implementation
Step 8: Implementation
A. Optimizing, suboptimizing, compromising; B. Legitimizing and consensus; C. Experts and expertise.
Step 9: Control of systems
Step 10: Evaluation of outputs, auditing and reappraisal

Furthermore, it would be wise to consider some of the laws and principles of GST in the design process. For instance, Yourdon, in his rather important work, developed the following four GST-based principles and applied them in information systems design (26). *Principle 1*: The more specialized or complex a system, the less adaptable it is to a changing environment; *Principle 2*: The larger the system, the more resources are required to support it, with the increase being nonlinear rather than linear; *Principle 3*: Systems often contain other systems or are themselves components of larger systems; *Principle 4*: Systems grow over time, both in terms of size as well as structural complexity. Yourdon's work is generally considered to be successful. Later, Caddy and Helou applied these principles to supply chains and supply chain management (27). The authors claim that application of these principles provided a deeper insight into these systems, and helped to develop better supply chain management processes. The authors of the present paper believe that the designers of automation/control systems need to keep the following laws and principles of GST in mind throughout the design process.

1. *The complementary law*: Any two different perspectives (or models) about a system will reveal truths regarding that system that are neither entirely independent nor entirely compatible.
2. *The law of requisite variety*: Control can be obtained only if the variety of the controller is at least as great as the variety of the situation to be controlled.
3. *The law of requisite hierarchy*: The weaker and more uncertain the regularity capability, the more hierarchy is needed in the organization of regulation and control to get the same result.
4. *The law of requisite parsimony*: Human short-term memory is incapable of recalling more than seven plus minus two items. Three elements and the four interacting combinations of them will consist of such seven items.
5. *Darkness principle*: No system can be known completely.
6. *Redundancy of potential command principle*: In any complex decision network, the potential to act effectively is conferred by an adequate concatenation of information.
7. *Steady-state principle:* For a system to be in a state of equilibrium, all subsystems must be in equilibrium. All subsystems being in a state of equilibrium, the system must be in equilibrium.
8. *Self-organizing systems principle*: Complex systems organize themselves and their characteristic structural and behavioral patterns are mainly a result of interaction between subsystems.
9. *Basin of stability principle*: Complex systems have basins of stability separated by thresholds of instability. A system dwelling on a ridge will suddenly return to the state in a basin.
10. *Viability principle*: Viability is a function of the proper balance between autonomy of subsystems and their integration within the whole system, or the balance between stability and adaptation.

Difficulties in control of complex systems are primarily related to the following factors: (a) unexpected events taking place; (b) operator not having sufficient time to perform the task; (c) lack or insufficient knowledge of processes and the system; (d) insufficient or lack of readiness or preparedness; (e) insufficient or lack of resources (3). The task-controller relationships in the design process can be structured via two approaches: *Designing for Simplicity,* and *Designing for Complexity.* The former is based on reducing the demands on tasks or increasing the controller capacity, or doing both. Although it seems that it is possible to handle the system complexity by reducing the mismatch between the demand and capacity, the resulting system will have a built-in limitation. This is due to what is known as *the n+1* *fallacy*: the system is designed to handle *n* number of possible states, but there is always the state *n+1* that has not been accounted for. *Designing for Complexity,* on the other hand, is based on the premises that complexity cannot be reduced to an arbitrary low level. In other words, the *Law of Requisite Variety* should be satisfied; that is the controller or operator should have at least as much variety as the system to be controlled. Since the designer can not reduce the requisite variety through interface design, she/he has no choice but to increase the variety of the controller. The resulting system is bound to perform relatively better since complexity of the reality is acknowledged rather than simplified. In fact, experiences seem to indicate that designing for simplicity is possible if one can transform the complexity involved according to a set of well defined rules, which is hardly the case in the real world. Hollnagel and Woods suggest the following: '*rather than designing for a simple world that does not exist, the goal should be to design for the complex world that does.'* (3).

4.2 The Life-cycle model suggested

In the CSE framework, system description is given in terms of goals and functions required rather than system components and capabilities. The designer needs, and asses the impact of changes on the roles of people in the system throughout the design process. Design involves assigning tasks and functions to humans and machines in the system, normally in a top-down manner. In 'conventional systems engineering', this is achieved through the development a functional architecture. This approach may be employed in CSE design as well, but the goals-means approach, as an alternative, may be preferred. The goals-means decomposition can be performed recursively; that is the means at one level becomes the goals at the next level down, etc. (3).

The Defense Acquisition Management Life-Cycle model adopted for structuring the design process in the framework is shown in Figure 2. System acquisition proceeds through six phases. In the *concept generation* phase, the aim is to generate and compare material and non-materiel solutions. Here, knowledge from operational experts and documents are gathered, and a ranking of potential approaches are done, and a capability analysis plan is prepared. The work involved in this phase has the following important cognitive aspects: acquisition of operational knowledge and its interpretation; supporting global assessments of human-related aspects of material and non-material approaches; advising on human issues that should be noted during concept comparison; advising on cognitive strategies for qualitative assessments; assessing competing concepts in relation to sociological, political, and cultural constraints within which the work will be executed (25). In *capability analysis*, the initial concept is refined, producing the following outputs: (i) a plan for technology development strategy; (ii) a systems engineering plan. This phase involves the following: specifying the required human functionality - integrated with technological functionality, and aligning it with operational demands; assessing the cost and risk, implications of staffing; changing the concept, if necessary. The primary goal of *technology development* is to reduce technology risk, where the primary tasks are preparing the following: a system development plan; an information support plan; a human systems integration plan; a staffing and manpower estimate report; a training development plan. In the *system development and demonstration* phase, the main goal is to reduce the risks that occur during systems integration and manufacturing, to ensure supportability during operations, and to integrate technology with human systems. Important products of this phase are a *system production and deployment plan*, a *production and evaluation plan*, and a *training plan*. The primary goal of *production and deployment* is to develop an operational capability to satisfy the needs of the mission. The products of this phase are: a deployment review; an operations, support, and retirement plan; a post-implementation review plan; an operations and support logistics plan. The *operations, support, and retirement phase* involves launching and sustaining the system and, eventually disposing it. The important products of this phase are *upgrade plans and a* *retirement plan*. The purpose of *Milestone A* is to authorize transition of the design into technology development; of *Milestone B*, to authorize transition of the prototype into system development and demonstration; and of *Milestone C*, to authorize transition of the final design into production or procurement. The design team needs to adapt modeling, design, and evaluation strategies throughout the phases of system acquisition to accommodate the changing goals for the systems engineering effort.

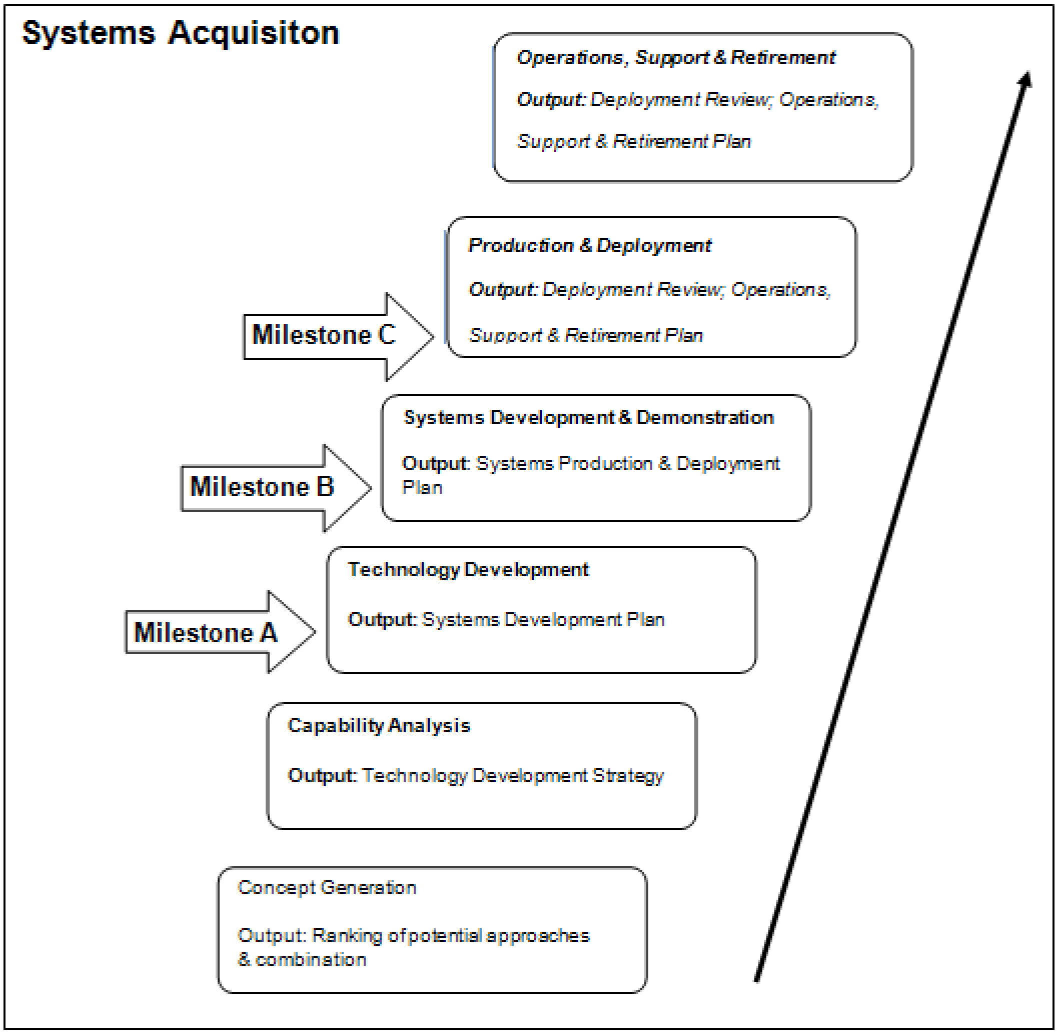

Figure 2 The Defense Acquisition Management Life-Cycle Model

The major contribution of cognitive systems engineers in this model would to work with program management and make sure that interdisciplinary participation is maintained throughout the design. They can play a significant role in knowledge elicitation and establishing communication between different disciplines. Cognitive systems engineers help multidisciplinary design teams move beyond a superficial understanding of the work system; they are equipped with methods and tools to identify and accommodate human needs, expertise, cognitive demands, constraints, and goals throughout the design process. Also, they can facilitate designs that posses capabilities of identifying, judging, attending, perceiving, remembering, reasoning, deciding, problem solving, planning, and other cognitive demands. Cognitive engineers can and should work with system engineers more closely on all aspects of design, including requirements, models and simulations, function analysis and allocation; they have a relatively better understanding of programmatic demands and constraints. Consequently, the chances of coming up with a final design with a well integrated perspective of multidisciplinary expertise may be increased. According to Deal, cognitive engineering is already embedded in systems engineering practice (28). He believes that systems engineers can work together with cognitive engineers, just as they work with practitioners of other engineering disciplines to improve the effectiveness of the systems created.

4.3 Developing the Control Hierarchy

The ECOM (The Extended Control Model) developed by Hollnagel and Woods (3) is shown in Figure 3. This model basically provides means of describing how a joint cognitive system (JCS) can perform on several layers of control in a hierarchical system. The model is made up of several concurrent control loops, some of which are closed-loop or reactive type, others are open-loop or proactive type, and some others are mixed. The *tracking* at the low end includes the activities required to keep the JCS within specified performance boundaries – these boundaries may be related to efficiency, safety, etc. The goals and criteria for the activities involved in this layer are determined by the regulating layer. The activities here are primarily closed-loop type. *Regulating* is also basically a closed-loop activity, but may involve some anticipatory control. The activities at this layer may not take place automatically, therefore may require some attention and effort. The plans and objectives for this layer are provided by the monitoring layer. However, the goals and plans provided may be changed, depending on the circumstances. At the *monitoring* layer, the objectives are set and plans are activated into actions. The *targeting* layer is concerned with targeting or goal setting. Some sub goals and activities may be generated too, following the goal-setting procedure; some of these may be automated or supported by information systems, while others may be related to performance criteria. The goal-setting activity is definitely open-loop type, implemented by a nontrivial set of actions, and often covering an extended period of time.

Several different control structures may be considered during the development of the control hierarchy given above. Most of these structures can be based on the general control system structure shown in Figure 4 (2). This structure is quite general, providing a GST/Cybernetics point of view, and can be adopted at different levels of the control hierarchy for different purposes. In this basic control cycle, the *receptor* (or *sensor* or *detector*) registers various stimuli. After its conversion into information, it is sent to the *controller* unit. The *comparator* (or *discriminator*) compares this value with a desired standard, and the difference, being a corrective message, is implemented by the *effectors* (or *activator*). Through monitoring and response feedback to the receptor, self-regulation is achieved. The controller may take a more sophisticated role when it includes a *goal-setter* with its *standard* reference, and a *decider* (or *selector*). Some controllers may also include a *designer* which formulates both the goals and the decision rules of the system. In all layers of the hierarchy, particularly at the higher levels, the design aims may require that the system has learning capabilities. The most significant advantages of living systems are considered to be *adaptation by learning* (2). Machines working according to cybernetic principles can also learn. Learning will take place if the general method and pattern of performance can be changed through the use of information flowing back from the system's performance. A general diagram of a learning system is shown in Figure 5. The information input to the system reaches the educable unit through the receptor where it is processed and goes through the effecter to become the system output. The decision unit compares the cause in the input with the effect of the output on the basis of the evaluation criteria stored in the comparator. The behavior of the decision unit is not predetermined; its internal parameters are modified, which results in learning or self-organization. It should be noted that all systems able to learn must necessarily organize themselves, but systems can organize themselves without learning. In a learning system, the rules must be adjusted in such a way that a successful behavior is reinforced, whereas an unsuccessful behavior results in modification. Here, managerial cybernetics may come to the designer's help; managerial cybernetics provides the guidelines for designing an appropriate organizational structure, including: (a) specification of the organization's sub-tasks and partition of work; (b) design of communication between subsystems; (c) definition of areas of decision-making and authority; (d) design and development of control systems and co-ordination of efforts toward the organizational goal.

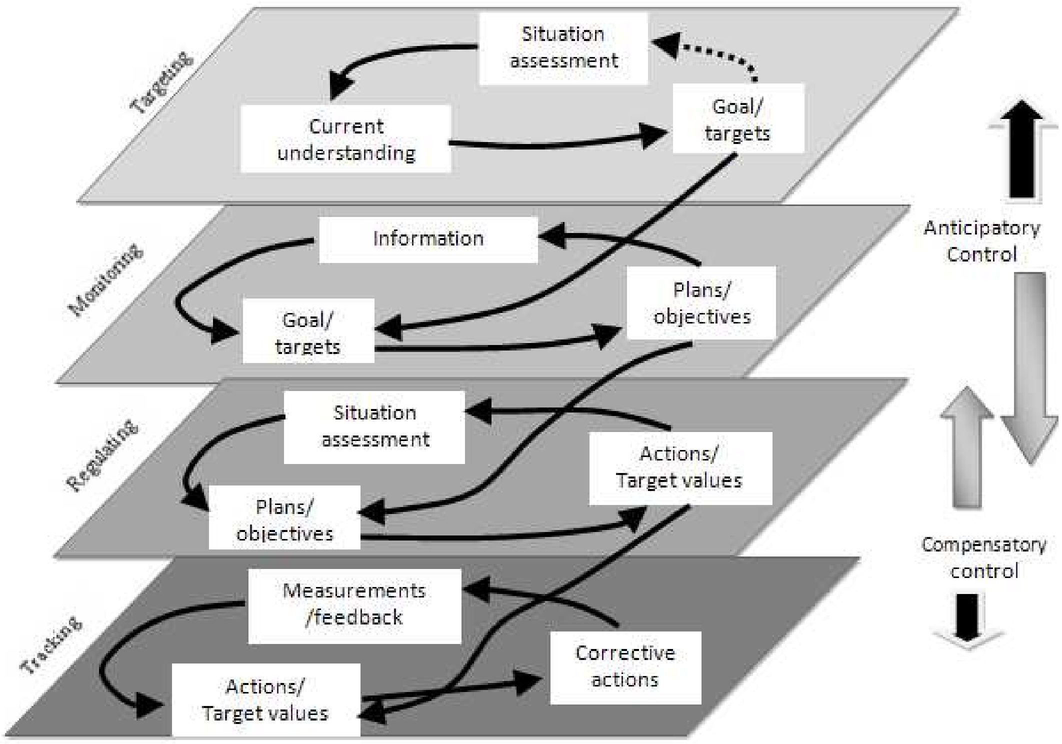

Figure 3 The Extended Control Model (ECOM).

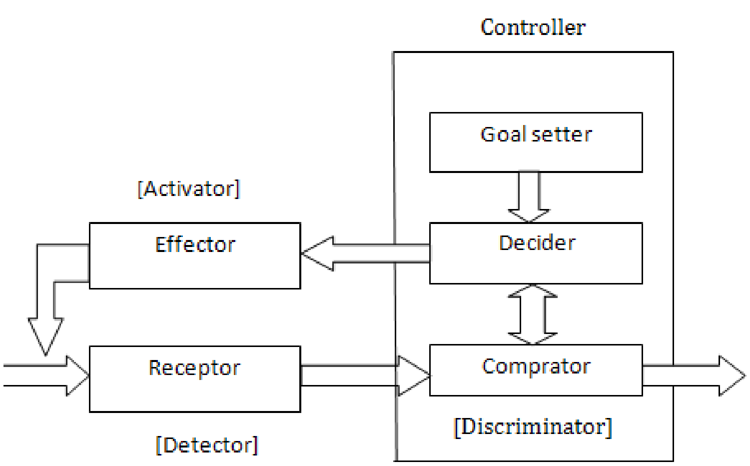

Figure 4 A general control system.

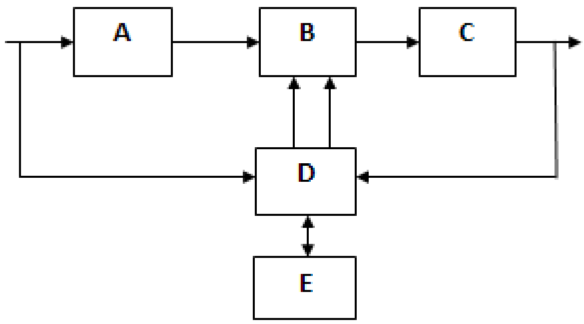

Figure 5 Diagram of a learning system.
A=Receptor; B=Educable decision unit; C=Effector; D=Comparator; E=Goal-Setter.

The automation and control hierarchy may also involve the use of decision support systems. The advances seen recently in computer and communication technologies have certainly increased the importance of decision making and decision support in complex systems, particularly at managerial levels. The classical view of decision support is based on normative decision making process within the framework of information processing paradigm. In the JCS paradigm, 'a more descriptive or naturalistic approach' is suggested (3). The implications of this view for design are as follows: (a) decision making is an overall function- intelligence must be viewed as a continuous process, not as a discrete one; (b) the important issue is to support monitoring, detection, and recovery rather than decision making; (c) decisions should not be automated; (d) if the environment is regular and predictable, there is no need for decision support; (e) decision support is needed when humans can not accomplish a task; (f) decision can not be separated from task intelligence, hence it must be present in the controller and in the implementation; (g) design of decision support needs to be based on an analysis of the control issues, particularly how control is lost and how it can be regained and maintained. It is fairly well known that even the most sophisticated AI-based intelligent systems or expert systems can not cope with complex reality; they reduce the complexity to their level. Hence, they can provide *only* partial answers to ill-defined problems (3), (5). Fully automated systems based on heuristic processes are known to be problematic. The reader here is reminded of Gödel's Incompleteness Theorem: 'for any formal system, there exits no decidable propositions which can not be proven by the formal system itself' (5). Furthermore, the meaning of symbols are context dependent, hence can not be formally described in principle - for instance, a system can not recognize an object for certain in computer-aided vision and pattern recognition. In short, extreme caution must be exercised in automating and in building decision support systems in human-machine systems. This caution is particularly important when designing high level and complex tasks; only human operators can handle certain complexities.

5. Conclusions

In this paper, a framework for designing complex automation and control systems is proposed. The framework developed is based on the important laws and principles of General Systems Theory. It involves a soft design methodology to be employed as the "philosophical" guide for design, The Defense Acquisition Management model to be used as the life-cycle model for structuring the design process, and the Joint Cognitive Systems Paradigm to be used in the development of the control Hierarchy. The General Systems Theory provides a strong theoretical context and some basic cybernetic structures to be employed in the design process. The proposed framework is soft in nature, and general enough to be adopted in diverse fields, including process industries, manufacturing industries, and service industries. Further research work is needed to provide a full integration of the methodologies and models involved in the framework through a real-life application.

6. References

(1) Parasuraman, R. and V. Riley, (1997), Humans and Automation: Use, misuse, diuse, abuse, *Human Factors*, 39, 230-253.
(2) Skyttner, L., (2001), *General Systems Theory: Ideas & Applications*, World Scientific.
(3) Hollnagel, E., D. D. Woods, (2005), *Joint Cognitive Systems: Foundations of Cognitive Systems Engineering*, CRC Press.
(4) Rousseau, D.M., (1977)," Technological Differences in Job Characteristics, Employee Satisfaction, and Motivation: A Synthesis of Job Design Research and Socio-technical Systems Theory", *Organizational Behavior and Human Performance*, 19, 18-42.
(5) Martin, T., J. Kivinen, J.E. Rijnsdorp, M.G. Rodd, W.B. Rouse, (1991), "Appropriate Automation-Integrating Technical, Human, Organizational, Economic and Cultural Factors", *Automatica*, Vol..27,901- 917.
(6) Jackson, M.C., (2000), *Systems Approaches to Management*, Kluwer.
(7) Jackson, M.C., (2003), *Systems Thinking: Creative Holism for Managers*, Wiley.
(8) Parnell, G.S., P.J. Driscoll, D.L. Henderson (Ed.), (2008), *Decision Making in Systems Engineering and Management*, Wiley.
(9) Hitchins, D.K., (2003), *Advanced Systems Thinking, Engineering, and Management*, Artech House.
(10) INCOSE Systems Engineering Vision 2020 INCOSE-TP-2004-004-02 September, 2007.
(11) Blanchard, B.S.,(2004), *Systems Engineering Management*, Wiley.
(12) Buede, D.M., (2000) *The Engineering Design of Systems: Models and Methods*, Wiley.
(13) Hazelrigg, G.A., (1996), *Systems Engineering: An approach to Information-Based Design*, Prentice Hall.
(14) Kossiakoff, A., W.N. Sweet, (2003), *Systems Engineering: Principles and Practices*, Wiley Interscience.
(15) Sage, A.P., (1992), *Systems Engineering*, Wiley.
(16) Sage, A.P., (1995), Systems *Management for Information Technologies and Software Engineering*, Wiley Interscience.
(17) Sage, A.P., J.E. Armstrong Jr., (2000), *Introduction to Systems Engineering,* Wiley.
(18) Stevens, R. P. Brook, K. Jackson, S. Arnold, (1998), *Systems Engineering: Coping with Complexity*, Prentice Hall - Europe.
(19) Maani, Kambiz E., R.Y. Cavana, (2007), *Systems Thinking and System Dynamics: Managing Change and Complexity*, Pearson - New Zealand.
(20) Checkland P., J. Scholes, (1990), *Soft Systems Methodology in Action*, Wiley.
(21) Checkland P., (1993) *Systems Thinking, Systems Practice*, Wiley.
(22) Flood, R.L., E.R. Carson, (1998), *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*, Plenum Press.
(23) Pidd, M. (Ed.), (204), *Systems modeling; theory and practice*, Wiley.
(24) Rasmussen J., A.M. Pejtersen, L.P. Goodstein (1994), *Cognitive Systems Engineering*, Wiley.
(25) Millitello, L.G., G. Lintem, C.O. Dominguez, "Cognitive Systems Engineering for System Design*", INCOSE INSIGHT, 12: 1, pp. 11-14, (2009).*
(26) Yourdon, E., (1989), Modern Structured Analysis, Yourdon Press, Prentice Hall International, Englewood Cliffs, New Jersey.
(27) Caddy, I.N., M.M. Helou, "Supply chains and their management: Application of general systems theory", *Journal of Retailing and Customer Services,* 2007, doi: 10.1016/j.jretconcer 2006.12.01.
(28) Deal, S., "You are a cognitive engineer? What is it you do", *INCOSE INSIGHT, 12: 1, p3, (2009).*