Interactions Between Systems Engineering and Human Engineering

October 28, 1999

Prepared by:
Melissa Dugger melissa_dugger@teambci.com
Chris Parker chris_parker@teambci.com
John Winters john_winters@teambci.com
Basic Commerce and Industries, Inc.
16347 Dahlgren Rd.
PO Box 1748
Dahlgren, VA 22448

John Lackie jhlackie@sonalysts.com
Sonalysts
PO Box 1839
Dahlgren, VA 22448

Contract No. N00178-96-D-2007
Delivery Order 0020

Table of Contents

Introduction

This paper describes significant interactions that occur between human engineers and systems engineers during system development. These interactions include information that must be shared, decisions that must be made, and actions or decisions that require approval. The interactions were identified based on sets of operational sequence diagrams (OSDs) of both systems engineering and human engineering. The OSD format used to document these processes is a modification of the methodology and symbology of traditional OSDs. OSDs are typically scenario-based and used for detailed man-machine interface analysis. They are used to illustrate the transfer of information and order of activities and to document allocation decisions and assumptions. The methodology used in this instance is intended to support higher-level descriptions of system operation and processes. The focus is on documenting the overall capability rather than a specific scenario.

The systems engineering and human engineering OSD sets were documented to support the development of the Manning Affordability Initiative's Human Centered Design Environment (HCDE). The HCDE is intended to be a prototype for a collaborative design environment that supports the consideration and inclusion of human operators and users throughout the design process. The OSDs break the process into task units, which typically include associated information requirements, decisions, and products. The diagrams for systems engineering were developed first, and the human engineering diagrams show how human engineering is performed in the overall context of systems engineering or system development. The identification of common products, tasks, and information requirements permits the definition of interactions between the two processes.

An effort has been made to describe the interactions in a stand-alone manner that does not require familiarity with any specific systems engineering or human engineering process. However, it should be noted that the perspective taken is generally from the systems engineer's point of view. In the detailed interaction descriptions, the context of the interaction within systems engineering is described first, followed by a description of the manner in which there is interaction with the human engineer. Throughout the descriptions, the terms "systems engineer" and "human engineer" are used. Although these are the singular forms, the terms could equally be pluralized or described as engineering teams. Many definitions exist for what qualifies a person to be labeled a "systems engineer," but within the context of this report some specific criteria apply. The systems engineer is the individual who has responsibility for the design of the system as a whole. The systems engineer's role may include programmatic responsibilities, but the emphasis in these descriptions is the technical aspect. The systems engineer may have a very active role in the definition of requirements or system functions, but his or her responsibilities change during the physical design of the system. At this point the purpose of the systems engineer is that of an integrator, and he or she is responsible for combining and deconflicting proposed designs submitted by engineers who specialize in particular disciplines or are responsible for particular subsystems. The human engineer plays one of these roles. The human engineer specializes in job and task design and the interaction of humans with one another and with automation, and his or her responsibility covers the human subsystems within the system to be designed.

The first section of this report briefly discusses what appear to be the most significant ways in which systems engineers and human engineers can interact. An interaction is considered significant if it has a relatively large impact on the design of the overall system or if its omission can lead to a drastic redesign of the system. An interaction could also be significant if the proper execution of the development process requires a great deal of iteration or communication between the systems engineer and the human engineer.

The main section of this report describes the interactions in greater detail. The list of interactions has been grouped into eight major categories that are roughly ordered according to standard phases of the systems engineering process. These systems engineering phases include:

1. *Mission Analysis*
2. *Requirements Analysis*
3. *Function Analysis*
4. *Function Allocation*
5. *Task Design and Analysis*
6. *Human Interface and Team Development*
7. *Performance, Workload, and Training Level Estimation*
8. *User and Requirements Review*

The description of each interaction includes a summary of how these interactions relate to the systems engineering processes described in the previously mentioned systems engineering OSDs and in IEEE 1220-1998, the Standard for Application and Management of the Systems Engineering Process.

Significant Interactions

Based on the interactions described in this report, four overarching interactions or themes have been selected as significant.

- Scenario Definition and User Review
- Participation in Function Analysis
- Function Allocation Decisions
- Compatibility of Models

These interactions are not meant to represent the bulk of the human engineer's work; they are intended to represent the most important ways in which the human engineer must interact with the systems engineer or other designers. The interactions do not necessarily represent what is currently planned or carried out in system development, but they instead represent key interactions through which human engineering can be better integrated within systems engineering.

More information about these interactions is available in the main section of the report.

Scenario Definition and User Review

The human engineer is often required to extend previous scenarios or build new scenarios that provide details about how the operators and users interact with the rest of the system. Different phases of operation can be described, and scenarios may cover both typical conditions and worst-case situations. The scenarios are used to build task and job analyses for the operators and users and to test designs. Since these scenarios are written from the perspective of the users and operators, they can be excellent vehicles for soliciting feedback during user reviews. Scenarios can be simply represented as written descriptions or storyboard sequences and therefore, they can be used in early stages of system development. The detailed inner workings of the hardware and software do not need to be defined because such details are irrelevant from the user's perspective. Reviewers such as potential users typically are able to provide better and more detailed feedback from a descriptive scenario than from a list of requirements or functional description. The review of the scenarios, however, can provide feedback on the system's physical design, functional capabilities, or even performance requirements. Without some sort of review, the system engineer can only assume that the system's requirements are compatible with the needs and limitations of the users or operators.

The human engineer is typically the designer who is best suited to perform user reviews. To allow scenarios to be used in this way, the human engineer must have scenarios that accurately represent the operation of the system. The human engineer must also be prepared to collect feedback on issues such as requirements and system functions instead of only control and display configurations. With adequate interaction between the human engineer and the systems engineer, scenarios and user reviews can allow for early and rapid feedback on system requirements, functions, and designs.

Participation in Function Analysis

Since function analysis is largely performed without regard to the allocation of the system's functions, it may be seen as an area that requires little if any human engineering participation. There are, however, two distinct reasons for human engineering participation that can reduce the potential for having to change the function analysis at a later date. First, the human engineer can assist in identifying functions that must be included because of the presence of humans within the system. Some functions may be required regardless of the humans' assigned responsibilities. Other functions will become apparent once some preliminary allocations are made, including those allocations that may be assumed from the system's initial concept of operations. Second, much of the human engineer's later work in task design and analysis will be driven by the results of the function analysis. Any information on the timing, sequence, or interaction of functions can be highly useful in the design of human tasks and jobs. Without human engineering participation, the function analysis is likely to contain insufficient details for functions to be allocated to humans. The human engineer is then left to make assumptions about the information or to continue the function analysis.

Function Allocation Decisions

Since accurate allocation of functions requires the consideration of the capabilities and limitations of humans, the participation of the human engineer is obviously required. The human engineer can provide reasonable estimations of what functions should and should not be allocated to humans. The systems engineer and other participants in the function allocation process are likely to have a good idea of the capabilities and limitations of humans in general, but the human engineer is likely to know more about the specific capabilities and limitations of the intended users. The earlier this participation occurs, the better the result is likely to be. The human engineer can assist in identifying some functions that must have a particular allocation. Reasons for such decisions include functions that are absolutely beyond the capabilities of the anticipated users, assumptions made as part of the system's initial concept, and grouping of functions that will benefit job design. Making these mandatory allocations as early as possible helps define the system to greater detail and also prevents these allocations from being made to the wrong system component.

Compatibility of Models

Proposed designs of systems, subsystems, or components can be evaluated before the system is constructed through accurate modeling. Although often limited in comparison with models of other disciplines, human engineering models can provide useful information about how humans interact with one another or with the rest of the system. Such models can help the human engineer optimize the performance of humans within the system. The main goal of the human engineer, however, should be to set the performance of the human to optimize the performance of the overall system. In order to accomplish this, the human engineering models need to be compatible with other models used in the design of the system. Without such compatibility, the human engineering models will not include an accurate representation of the system's hardware and software. Model compatibility is required for the human engineer to produce accurate models of human performance and to be able to model how human performance impacts the performance of the overall system.

Detailed Interaction Descriptions

This section of the paper outlines all of the systems and human engineering interactions uncovered from task analyses of the two processes. These descriptions are listed in the order of the systems engineering phases. Each interaction begins with contextual information to characterize the design process at the time of the interaction. Additional detailed information about the interaction follows, as well as the implications for the process. Finally, references to IEEE 1220-1998 and the Systems Engineering OSDs (SE OSDs) are provided.

1 Mission Analysis

The mission analysis phase of system development includes the determination of the overall system capabilities and the system's mission or purpose. Scenarios or mission profiles are created. The boundaries of the system need to be identified, as do the interactions of the system with its environment and with other external systems.

1.1 Selection of Comparison Systems

A frequently used approach in system development is comparison of the system under design to predecessor systems. While this technique is more straightforward for evolutionary designs, it may still be employed for systems that have no direct predecessor. All or part of the current system may be compared to all or part of some previous system that served a similar function, had a similar goal, or included similar components. This may be a formal process in which the performance and attributes of the predecessor system are quantified and set as a baseline upon which the new system must improve. It could also include a review of lessons learned from previous systems. Although it may be informal or even unintentional, some comparison is performed any time the developers have prior experience with the development or use of similar systems.

Within the human engineering process, previously designed or built systems or subsystems are selected for comparison with the system under design. The system under development may have multiple comparison systems or a variety of comparison subsystems from different pre-existing systems. The human engineering practitioner may observe or otherwise analyze the performance of the comparison systems to establish design goals or performance requirements. Among the different types of data that may be collected are historical data, observational data, user data or feedback, and data from experimental prototypes. Information on past performance of multiple comparison systems may be used to select or narrow options for designs.

While the comparison systems must be similar to the current system in either mission or implementation, a system that is useful to the human engineer may not be useful at the overall system integration level. The human engineer, however, will be required to address systems selected by the systems engineers or others as a baseline for comparison. Systems or subsystems that the systems engineer considers relevant for the human engineer should be assessed by the human engineer to confirm their similarity and applicability to the system under design. The human engineer may find information on comparison systems selected by the systems engineers to be very useful. The human engineer may need approval for the use of some comparison systems identified for system components under human engineering design responsibility. An early identification of comparison systems will allow the subsequent recommendations to have a more effective influence on design decisions.

IEEE 1220-1998: 6.1.2 – Define project and enterprise constraints
6.1.3 – Define external constraints
SE OSDs: SE110 – Define and Assess Operational Environment

1.2 System Use Scenarios

Tools such as system scenarios, design reference missions, and mission profiles or timelines are used by a variety of disciplines during system design. Information from these sources can be used to identify required interactions with external systems, determine functional requirements for a system, and establish performance requirements for interaction with external systems. Once designs are complete, such scenarios and timelines may be used to evaluate or validate system design options.

In order to adequately account for the users or operators of the system under development, some scenarios must be defined from their perspective. System use scenarios describe, from the user's point of view, detailed events of the system mission, including identification of mission phases, mission time scale, and events external to (and their interactions with) the system. Scenarios selected by the human engineer should address both cognitive and physical tasks and should emphasize impacts on human performance, potential environmental effects, and safety. These scenarios are required to perform job or task design, and they can be used to determine requirements for human-system interfaces. Scenarios from the user's perspective are powerful tools for eliciting user or subject matter expert feedback early in the design process.

System use scenarios defined by the human engineer will often be extensions or subsets of scenarios developed or approved by the systems engineers. The definition of system use scenarios will typically require assumptions on the part of the human engineer that further define the system. These scenarios, therefore, should be either approved or at least reviewed by the systems engineers. The human engineer must ensure that the scenarios accurately reflect a potential or achievable design.

The development and subsequent use of system use scenarios is critical for the human engineer. Without valid scenarios, it will be difficult if not impossible to account for users and operators in the design process. As scenarios extend assumptions about system design, those assumptions must be verified or accepted by other disciplines. Collaboration with the systems engineer in scenario development will increase the probability that suggestions from user or subject matter expert reviewers will be accepted. Tasks that are specific to the use of interfaces or team interaction will have to be added to system use scenarios, and these extensions will also need to be verified with the systems engineer and other design disciplines.

IEEE 1220-1998: 6.1.4 – Define operational scenarios
SE OSDs: SE110 – Define and Assess Operational Environment

1.3 User Environment Characteristics and Effects

The design of the system must account for the environmental conditions under which the system will be employed. A wide range of environments is possible, and all relevant factors should be considered. Natural conditions such as weather, topology, time of day, and lighting conditions are of interest, as are conditions such as noise and vibration induced by the operation of the system. Once the conditions are identified, the effects of those conditions and any resultant design constraints should be ascertained.

The human engineer will need to assess the environmental conditions catalogued by the systems engineers and determine whether or not all conditions that significantly affect humans have been identified. Operators and users must be shielded entirely from some environmental characteristics and other characteristics will influence their performance. The human engineer will need to quantify the effects of environmental characteristics on human performance and will provide the data to the systems engineers and other design disciplines for use in design decisions. In some cases, the human engineer will need to determine how to mitigate, eliminate, or compensate for environmental effects. As more of the system's physical design is completed, additional induced environmental factors will become apparent or better defined. The human engineer must therefore iteratively review system designs to continue to identify induced factors and determine how external environmental factors may affect humans. In some cases, the human engineer will make assumptions about environmental factors that are present and will need to clarify or present those assumptions to the systems engineers.

Once the effects of environmental factors have been assessed, it must be determined whether or not desired levels of system and human performance can be achieved. In some cases, the performance effects of the environment will need to be included in system or component models and simulations. The systems engineers and other designers will need to know about such performance degradations, and will also need to be given specific requirements for equipment to mitigate or inhibit environmental effects that have been identified. Approaches to mitigate environmental effects include breathing or life support apparatuses, vibration damping, noise cancellation, hearing protection, lighting, and operator exposure or duty limits.

IEEE 1220-1998: 6.1.8 – Define utilization environments
SE OSDs: SE110 – Define and Assess Operational Environment

2 Requirements Analysis

During requirements analysis, source requirements are identified, clarified, and prioritized. The requirements are broken down or decomposed into greater detail. Each lower-level requirement must be traceable to higher-level requirements. As the requirements are defined in greater detail, they will become more specific to the planned implementation of the system, and the involvement of designers within different disciplines becomes necessary.

2.1 Human Engineering Constraints

Constraints are implied requirements that restrict the design of a system. They are not created directly from a specification, but they are instead the result of external limitations. Many more constraints will be involved in the development of systems that retain major components of previous systems. If more constraints are known early in the design process, it is easier to narrow the design space for the system.

Constraints that impact the work of the systems engineer are likely to impact the work of the human engineer as well. The overall constraints of the system should therefore be documented to ensure that all participants are aware of the restrictions. Some additional constraints will arise due to design decisions or analyses by the human engineer. Many constraints will come from the inherent limitations of humans in general. Once the characteristics of the user population become more certain, other constraints may become apparent. As they arise, these constraints must be identified and passed on to other design disciplines. In some cases, constraints from different disciplines must be developed and documented in parallel, requiring collaboration between design disciplines.

IEEE 1220-1998: 6.1.2 – Define project and enterprise constraints
6.1.3 – Define external constraints
SE OSDs: SE130 – Identify Constraints and Analyze Operational Requirements

2.2 Human Performance Requirements and Human Engineering Design Requirements

Much of the early work in developing a system involves the definition and decomposition of requirements. Requirements from a variety of sources and disciplines must be analyzed to remove conflicts. The human engineer is primarily responsible for two types of requirements, human performance requirements and human engineering design requirements. Human performance requirements include times and accuracies for tasks assigned to humans. The human engineer must ensure that the proposed requirements are in fact achievable by the intended operators and users. The human engineer may in some cases define the human performance requirements based on external requirements, specifications of other system components, or the capabilities and limitations of the prospective operators and users. The human engineering design requirements concern specific aspects of the hardware and software that are necessary to fit the operators and assist them in their assigned tasks. These requirements define what must be designed and constructed to permit the operators and users to interact with one another and the rest of the system.

Human performance requirements are frequently derived from or at least bounded by other performance requirements within the system. The accuracy, response time, and other attributes of the operator tasks will affect similar attributes at the system level. The requirements produced by the human engineer should therefore be in a format similar to that of the system-level requirements. Common format, both visually and electronically, will make the derivation of human performance requirements easier, and it will also make the verification or approval of those requirements by the systems engineers a simpler task. In the same way, the human engineering design requirements should share a common format. In the case of these requirements, a common format is even more important as they must be reviewed or followed by system designers in other disciplines. Although the human engineer is the one who may set specifications for the design of other system components, the complete design and construction of those components will be the responsibility of others within the project. As designs become more detailed, a continuous interaction between the human engineer and other disciplines becomes more advantageous. The implementation of the requirements needs to be verified, and additional design decisions need to be made as the design progresses.

IEEE 1220-1998: 6.1.11 – Define performance requirements
6.1.14 – Define design characteristics
SE OSDs: SE130 – Identify Constraints and Analyze Operational Requirements
SE140 – Identify Functional and Performance Requirements

3 Function Analysis

Function analysis involves the translation of the system's requirements into a functional architecture that defines how the system will meet those requirements. The functional architecture does not include references to allocation or implementation, but some functions will be included because of implementation decisions. The process of function analysis includes the decomposition of top-level system functions into greater levels of detail and the definition of how the functions interact.

3.1 Functional Decomposition

A high-level set of desired system functions is typically specified very early in the development of a system. These top-level functions must then be broken down into their component subfunctions that meet the system's requirements within the specified constraints. Once the functions have been defined, they can be allocated to be performed by humans, hardware, software, or combinations. A single function can often be decomposed in a variety of ways. Choosing the best decomposition before function allocation decisions are made will reduce later design changes.

Decisions on function allocation are typically made iteratively as functional decomposition continues. Allocating the functions permits their parameters to be specified in greater detail and serves to verify the decomposition. Decomposition of the functions must continue as the attributes of the subfunctions are needed to support design decisions. Although the definition and decomposition of functions is independent of allocation and may be seen as not relevant to the human engineer, the results of the decomposition and analysis will be used in later design work. Much of the information that is critical to the human engineer may not be of interest to those performing the decomposition. Timing requirements, available information, required information, and other inputs may be necessary for subsequent human engineering design decisions. The optimal way to ensure that the necessary information is defined is to have the human engineer work in conjunction with other designers. This collaboration will allow the definition of function parameters required for the work of the human engineer. The alternative is to wait until the human engineer needs additional information and either request the necessary information or generate it at that point. Any new functional information that the human engineer independently generates will need to be reviewed and verified by other designers.

IEEE 1220-1998 6.3.2 – Functional decomposition
SE OSDs: SE210 – Functional Definition

3.2 Review of Functional Architecture

The functional architecture of a system represents, without specifying allocations, how the system will meet its requirements. The architecture includes the required functions, the flow and timing between functions, and their respective inputs and outputs. As with functional decomposition, the functional architecture is highly relevant to the human engineer despite the fact that it does not explicitly include any allocation decisions. The functional architecture does, however, imply some allocation decisions. Some functions are required in order to support specific implementation options. A system with a nuclear power source will have a refueling function just as other implementations do, but the timing of the function is likely to be longer in both duration and periodicity. If humans are to be included as part of a system, functions such as life support, food supply, and decision support are relevant.

It is the human engineer's responsibility to review the functional architecture and ensure that it includes all aspects relevant to the inclusion of humans in the system and their projected roles. In the case of top-level system requirements, the human engineer can provide feedback as to whether or not additional high-level functions need to be added to account for the role of humans proposed in the system concept. While it is likely that few if any functions are added at this level, additional functions may be catalogued for inclusion during functional decomposition. The functional flow of the system needs to be assessed to ensure that it is compatible with the inclusion of humans in the system. Enhanced analysis is possible as more allocation decisions are made and as greater levels of decomposition are reached. An additional way that the human engineer may use the functional architecture is to attempt to identify new requirements that surface due to the inclusion of humans within the system. The functional architecture needs to be compared to the human engineering requirements, specifically human performance requirements, to determine whether or not they are satisfied by the functional architecture.

IEEE 1220-1998: 6.3.3 – Establish functional architecture
SE OSDs: SE210 – Functional Definition

4 Function Allocation

The goal of function allocation is to effectively distribute the functions of the system between humans and technology. The functional elements are identified and then utilized in the creation of functional element allocation options. In developing these allocation options the systems engineer considers the project constraints, requirements, and the capabilities and limitations of both technology and the operators. The constraints and requirements to be considered are usually developed early in the overall process when the systems engineer is assessing all the constraints on the system and its operational requirements. The systems engineer determines the capabilities and limitations of the potential technologies, as well as the possible use of commercial off-the-shelf products, while information about operator capabilities and limitations will come from the human engineer. In addition, certain functions may be required to be allocated specifically to operators or technology. These allocations are made first, and then the remaining options are assessed and allocated. This mandatory function allocation, as well as the development of functional element allocation options, is an important step in the systems engineer’s creation of candidate physical architectures for the system.

Much of the function allocation responsibility falls into the realm of human engineering. One way for the human engineer to go about this task is to identify the capabilities and limitations of both the potential operators and human engineering technologies and then weigh the various options to determine possible allocations. The human engineer first determines which functions must be allocated specifically to a human or machine, and then conducts the tradeoffs to develop additional potential allocations. The mandatory and additional allocation recommendations are preferably co-developed by the human engineer and system engineer, or developed independently by the human engineer. The system engineer must then approve the recommendations.

4.1 Consideration of Human Engineering Technologies

In order to make the best decisions about which functions should be allocated to technology, it is important to be aware of the types of technology available and their inherent capabilities and limitations. The systems engineer conducts studies to assess the general capabilities and limitations of the technology available that may be useful for the particular system under design.

The human engineer conducts additional research and analysis to identify the technologies specifically applicable to human engineering and then further defines their capabilities and limitations. Relevant technologies include decision support systems, human performance models, and human-computer interaction techniques. An accurate assessment of the potential human engineering technology allows the human engineer to tradeoff these factors with the capabilities and limitations of the operator. The human engineer's identification of the human engineering technologies and assessment of their capabilities and limitations should be done with the help of other disciplines to avoid duplication of work and ensure common assumptions.

The human engineer can eliminate redundant work by consulting with the systems engineer and making use of the previous systems engineering studies of technology capabilities and limitations. Additionally, the human engineer can aid the systems engineer by providing necessary data about the operators. The human engineer will consider the future operators and assess operator capabilities and limitations. These capabilities and limitations will be important factors in the human engineer's function allocation, but will also be needed by the systems engineer to assess the capabilities and limitations of the system as a whole.

IEEE 1220-1998: 6.5.5 – Assess technology requirements
SE OSDs: SE310 – Synthesize Multiple Physical Architectures

4.2 Early Identification of Mandatory Allocations

One of the first steps in allocation is the identification of functions that must be allocated specifically to a human or a particular technology. For example, if there is a complicated numerical calculation that must be completed very quickly, this should probably be allocated to software. On the other hand, if there is an important decision that must be made, such as whether or not to fire on a potential enemy, it may be determined that this function should not be left to a machine but should be the sole responsibility of an operator. The systems engineer will make these mandatory allocation decisions, based in part on recommendations from the human engineer.

There are a number of information sources that might be important for the human engineer to consider while developing mandatory allocation decisions.

Information external to the design may include documents such as the Concept of Operations or human engineering literature applicable to the design domain. Sources of information from within the human engineering process that might be useful are the system use scenarios or the variety of documents outlining requirements, constraints, and capabilities/limitations.

The systems engineer will work with the human engineer and provide him with a variety of information sources developed by the systems engineering team, including the list of functional elements, draft functional architectures, and cost constraints. The systems engineer and human engineer should also consider if there are additional technologies that are available or expected to be available that should be investigated for an optimal allocation. If so, systems engineering trade studies might be conducted to assess the options and the results of these studies would be shared with the human engineer.

The development and approval of the recommendations for the mandatory design allocation follows the general process for allocation recommendations (as outlined below). However, it is important to note that the human engineer should consider the mandatory allocations early in the design process and present this information to the systems engineer. If the mandatory allocation decisions are finalized early, this can prevent wasted effort on designs that do not match the mandatory requirements.

IEEE 1220-1998: 6.5.1 – Group and allocate functions
SE OSDs: SE310 – Synthesize Multiple Physical Architectures

4.3 Development and Approval of Function Allocation Recommendations

Both the mandatory function allocations and the additional allocations that follow must be developed by taking into account a number of factors and considering a variety of information from the systems engineering process, the human engineering process, and sources external to the design process. This can be a complicated step in the design where conflicting costs and benefits require careful tradeoffs. If the allocation decision is ambiguous, systems engineering trade studies or human engineering studies, such as user review or performance and workload estimation, may need to be performed. The allocation recommendations will ideally be generated jointly by the systems engineer and human engineer, but they may instead be developed independently by the human engineer and submitted to the systems engineer for approval. If the human engineer prepares the options, then the expectations of the systems engineer (i.e., number and variety of options desired) should be taken into account.

Once the recommendations are developed, they must be approved by the systems engineer. If the systems engineer was also involved in development, then the approval should be a simple step. However, if the human engineer developed the recommendations independently, the systems engineer may have feedback or suggestions for changes. In addition, the systems engineer should be aware of other influential decisions that might have been made or are being considered. Thus, the systems engineer should be able to take into account the objectives of the human engineer's suggested allocation and the objectives of the activities of other disciplines. This may be an iterative process of refinement until the systems engineer and human engineer can agree on a set of allocations.

IEEE 1220-1998: 6.5.1 – Group and allocate functions
SE OSDs: SE310 – Synthesize Multiple Physical Architectures

5 Task Design and Analysis

Once the functions of a system have been assigned to particular system components, the functions can typically be defined to greater resolution of detail. What was a generalized function must be expanded to describe exactly how the specified system components will accomplish the functions. Functions that have been allocated to humans are typically referred to as tasks. Given the constraints of the system's requirements and functional architecture, the human engineer needs to define precisely how the humans within the system will carry out their assigned tasks. The human tasks include both tasks that humans do alone and tasks that involve their interaction with other parts of the system. The order and interactions of the tasks can be defined and modeled to verify that they meet the system requirements.

5.1 Development of the Task List

Before the analysis of the tasks of the humans, it is necessary to compile a complete list of the tasks to be considered. This process may also include the decomposition of tasks, if such a decomposition would be useful. Most likely, the human engineer will be responsible for creating the task list; however, he or she may want to work with the systems engineer to achieve a better understanding of the tasks. Systems engineering documents, such as the functional element allocation options and the physical architecture, may be of great use to the human engineer. The systems engineer might also provide additional, amplifying information, such as decisions by other disciplines that influence the tasks of the humans.

The human engineer will assess the information from the systems engineer and other design engineers and devise a complete list of human tasks. Additional inputs to the development of the task list include the approved function allocations and interface-specific tasks, if applicable. Interface-specific tasks are those that are created as a function of the interface that is chosen, and are based on the interface concepts and designs. Interface-specific tasks are normally defined following task design; however, due to the iterative nature of the design process, the human engineer may redevelop the task list in light of later decisions.

IEEE 1220-1998: 6.5.2 – Identify design solution alternatives
SE OSDs: SE320 – Evaluate and Select Preferred Architecture

5.2 Identification of Task Characteristics, Interactions, and Sequences

Once the task list has been generated, the particular characteristics of each task must be outlined. This further definition facilitates a better understanding of the individual tasks and can be used in other steps of the task design and analysis process. In order to attain a full understanding of the tasks, the interactions between the tasks or the possible sequences in which they will be accomplished should also be identified. The identification of interactions and sequences among tasks is important both to better characterize the tasks themselves and also to create accurate task models.

The task design and analysis portion of the human engineering process might be highly iterative, and the results of both these identifications can act as inputs for each other. Additional information sources might include the human engineer's task list and the externally set operational requirements. Systems engineering contributions include the functional element allocation options and general systems engineering guidance on the current system design. This systems engineering advice is imperative in order to accurately identify interactions with non-human elements of the system. These task interactions include interactions between humans and automated functions and/or other system components.

There is also an interaction between the human engineer and systems engineer because the human engineer's task definition is dependent on the system design, since this design will impact the possible ways to accomplish the tasks. The human engineer can create the most useful set of task characteristics only by checking with the systems engineer to verify that the human engineer has a correct understanding of the system design. The most accurate representation of the system design is probably embodied in the systems engineer's current candidate physical architectures. The systems engineer's functional decomposition will also be useful to consider. If the decomposition is not to the level of detail required by the human engineer, a further functional analysis may be necessary.

IEEE 1220-1998: 6.5.2 – Identify design solution alternatives
SE OSDs: SE320 – Evaluate and Select Preferred Architecture

5.3 Selection of Modeling Tools and Techniques

Modeling techniques are typically used to evaluate or compare candidate designs. Models can be constructed at the functional level or in the context of the system's implementation. Creating models of external systems can help to define functional and performance requirements for the system under development. The utility of modeling techniques and executable models in particular can be significantly increased if models used by different designers are interoperable. Systems engineers can then create higher-level models of the system by combining models developed for different subsystems or within different disciplines.

An important step for the human engineer in task design and analysis is to select appropriate task-level tools and techniques that will result in a useful and appropriate model. The tools and techniques should be chosen early enough to ensure that they can support the inclusion of relevant information from the task analysis. These modeling tools and techniques will determine how the task list, task characteristics, and task interactions and sequences will be used to create task models. Comments from subject matter experts might also be useful in tailoring and validating task models. In order to have task models that are compatible with system-level models and models from other design disciplines, the human engineer and systems engineer must agree on the modeling tools to be utilized. The human engineer should also request the systems engineer's input on the models, since they will include non-human elements. Since model development can take considerable time and resources, communication between the systems engineer and human engineer is important to ensure that the models selected can be used across design disciplines. Given the importance of resource allocation to support system and subsystem modeling, overall project plans should include human engineering modeling as a programmed milestone.

IEEE 1220-1998: *6.5.2 – Identify design solution alternatives*
6.5.11 – Develop models and prototypes
SE OSDs: *SE320 – Evaluate and Select Preferred Architecture*
SE330 – Integrate System Physical Configuration

5.4 Task and Function Audit

In synthesizing the physical architecture, allocations between humans and machines will be reflected in the design of interfaces. The designers will have to verify that all functions in the functional architecture can be traced to tasks performed by either humans or automation. A review of the task list – including interface- and team-specific tasks – should therefore find all of the tasks drawn from the function allocation in the interface and team concepts and designs. This review may be thought of as an audit of the interfaces with a mandatory consideration of all of the tasks from the analyses and simulations.

A confirmation of automation assumptions by the systems engineer and other designers is necessary to ensure that the job and task design performed by the human engineer does not omit necessary functions. The human engineer may need to give feedback to the systems engineer about tasks that could be automated or tasks that need further design support. For example, further function analysis may be required, an operator may need additional information to support decision-making, or additional automation or system functionality may improve system performance.

IEEE 1220-1998: 6.5.15 – Finalize design
SE OSDs: SE320 – Evaluate and Select Preferred Architecture

6 Human Interface and Team Development

Designs and concepts for the interfaces between humans and software, hardware, and other humans need to be identified and developed. Three levels of interfaces are described starting with individual interfaces that represent a particular interaction based on the task analysis as well as performance and design requirements, then combinations of interfaces for a design at the individual operator level. These individuals are then assembled into crews or teams employing multiple operator interface designs and concepts. The creation of the separate levels of interfaces may be performed in any order depending on the availability of resources and the priority of individual user versus crew/team development.

6.1 Points of Human Interface

Points of human interface may be thought of as the content and the location (origin and destination) of information that may be conveyed between system components (specifically, between humans or between a human and a machine). Also included are the data to be transmitted, the nodes or elements between which the data is to be transmitted, when the data is transmitted and other interface-specific constraints, such as special conditions based on times and events. These points will be used in the development of the interface concepts and designs and will lead to interfaces at the individual level followed by the crew/team level.

The human engineer must identify all of the data to be transmitted and the location, or nodes, to and from which it will be transmitted. This is based on the functional decomposition and allocation, as well as the task analysis (which includes characteristics of tasks and the interactions and sequences), and any available internal and external interface information developed to that point by the systems engineer. These system-level interfaces must be decomposed for application to the level of automation.

The systems engineer helps verify allocation assumptions made by the human engineer and document the flow of information between humans and automation as well as identify additional points in the allocated functional architecture at which information or material is passed between humans and other system components. The role of the human engineer is to keep the points of interface in line with the initial system-level interfaces defined earlier in the systems engineering process and in line with the mission goals and constraints. Some interactions may be identified to address special needs or preferences of the users.

IEEE 1220-1998: 6.1.7 – Define interfaces
6.5.7 – Define physical interfaces
SE OSDs: SE210 – Functional Definition
SE320 – Evaluate and Select Preferred Architecture

6.2 Selection of Human Interface and Team Guidelines

For the development of interfaces and teams, human engineers need to be aware of any existing guidelines applicable to the information or material passed between humans or between humans and equipment. The guidelines will also assist in keeping the design in accordance with constraints, heuristics and prior research of the particular engineering or design community. Guideline topics may include, but are not limited to, short term and working memory limitations, display and control modalities, physical or strength limitations, and group dynamics. These guidelines may also include those defined in, derived from, or implied by human and job/task requirements and organizational design.

Collaboration between the systems engineer and human engineer on the selection and implementation of standards and guidelines will help identify how system-level guidelines may be applicable to human engineering designs. Full application of system-level guidelines will often require the implementation of specific, lower level, detailed guidelines. For example, if a particular computer system architecture is selected, then any associated user interface design guidelines should be implemented. Collaboration will also help identify how guidelines from one design discipline will impact other disciplines. The human engineer will identify additional useful guidelines, each of which may impact one or more other design disciplines.

IEEE 1220-1998: 6.1.3 – Define external constraints
SE OSDs: SE130 – Identify Constraints and Analyze Operational Requirements

6.3 Development of Interface and Team Concepts or Designs

Once an initial physical architecture has been synthesized and approved by the systems engineer, the interfaces between system components – such as humans, hardware, and software – can be developed. The interaction of humans with other system components will be based on the functional architecture, allocation decisions and human engineering inputs. Some elements of both internal and external interfaces will have already been defined as interfaces between functions within the functional architecture.

The human engineer will be responsible for designing and optimizing how individual humans interact with non-human system components and how humans act together as teams. Interface concepts and designs are developed based on requirements for interaction between humans and other system components specified earlier. Requirements such as the transfer of information, timing, and physical location must all be satisfied by the interface designs. Due to the potentially significant and varied amount of information to be transferred, the process of developing team and individual interface concepts and designs is highly creative. The concepts are less detailed and concrete than the designs but are highly iterative with their development, as they feed off of each other. The development of interfaces includes their physical appearance and procedures for use. Interface guidelines and standards will influence the design of the interfaces. Interfaces must be considered collectively, in combinations, in order to minimize conflicts between different interfaces encountered by a single operator or user. Team designs will be based on the allocation of tasks and other responsibilities to different operators or team members, and will be influenced by such factors as individual workload and performance levels, team design principles, and overall performance requirements.

Team and individual interface design will be highly constrained due to other design decisions, such as specific pieces or types of hardware and software that are to be used. The human engineer attempts to develop team and interface designs that provide for optimal system performance within those constraints. The human engineer requires input from the systems engineer on system-level constraints (particularly those imposed by other design decisions), project and enterprise constraints, off-the-shelf availability, make-or-buy alternatives, state-of-the-art capabilities, design solution alternatives, etc. In some cases, constraints and design decisions that have been made previously may need to be reevaluated based on analysis of human performance within those constraints as well as interaction with other design disciplines to ensure the feasibility of the proposed designs.

IEEE 1220-1998: 6.1.2 – Define project and enterprise constraints
6.1.3 – Define external constraints
6.1.7 – Define interfaces
6.5.7 – Define physical interfaces
SE OSDs: SE130 – Identify Constraints and Analyze Operational Requirements

7 Performance, Workload, and Training Level Estimation

The systems engineer must evaluate the design or design options proposed by system designers within the different disciplines. Evaluation of a single option is necessary to determine whether or not the system requirements are satisfied, and multiple options may be evaluated in order to make a selection. The systems engineer may determine which options meet requirements and then select the best alternative, or the best option may be selected and then compared to the requirements. Overall system performance is an important parameter, but it typically consists of multiple variables that may be measured within different design disciplines. The design evaluations provided by different disciplines will all need to be available to the systems engineer to enable the tradeoff of different design options.

To help in the evaluation of concepts and designs, the human engineer will estimate the physical and cognitive workload levels of individuals and teams within the system. Workload stressors and their effects on human performance and operator coping strategies, as well as the effects of task neglect or delay, need to be defined. Workload and the resultant manning and training requirements are to be optimized to meet required performance levels.

7.1 Individual and Team Workload and Performance Estimation

Workload levels can significantly influence the performance of many system components or subsystems, including humans. Once workload levels are predicted, performance measures can be adjusted to determine the impact of workload. Given the tasks allocated to humans, the human engineer needs to estimate the cognitive and physical workload demands of the tasks on the operators and users. Executable models or simulations are typically used, but subjective feedback from test users or subject matter experts may also be employed. Workload levels must be estimated for different scenarios or situations, and changes in workload level can be as important as the absolute levels of workload. Workload on the team as a whole, frequently quantified as the time required to complete all assigned tasks, also needs to be estimated. In order to be accurate, workload models need to include any operator or user tasks that are required to manipulate or utilize the human-machine interface.

To effectively estimate workload and performance, the human engineer needs up-to-date design data from the systems engineer and other designers. In order to create accurate models of how the humans interact with the rest of the system, the human engineer will need access to models of other system components. Without an accurate simulation of hardware and software functions and performance, the model of the human interactions will not be accurate. Information on other system components may be included as part of an executable model, or it may be used to create a physical prototype of portions of the system with which test users can interact. The true relevance of workload lies in its impact on human and system performance, not as a stand-alone measure, so workload measures should be easily integrated with performance models. Similarly, models of human performance need to be compatible with models that can predict overall system performance. The goal of the human engineer should not be to optimize human performance alone, but to put human performance within acceptable levels to optimize overall system performance. This goal cannot be accomplished without human workload and performance models that are compatible with higher-level system models. Model compatibility will also be important when design changes are made that necessitate alterations to the models.

IEEE 1220-1998: 6.5.11 – Develop models and prototypes

6.5.15 – Finalize design
SE OSDs: SE320 – Evaluate and Select Preferred Architecture

7.2 Training Concept Evaluation

The resources required to field and maintain a system are typically key concerns of the systems engineer. The overall cost of the system includes the cost to prepare it for use and to maintain it over its life cycle. If the human is considered to be part of the system, then the resources required to prepare and provide operators and users are just as relevant as the resources required to provide hardware and software. The users and operators are frequently the most often changed and varied parts of the system. The training required to prepare them for use of the system and to maintain their qualifications as users and operators are important parts of the system life cycle support requirements.

In the development of a particular system, training may or may not be considered to be part of the human engineer's responsibilities. Even if the human engineer is not directly responsible for developing training requirements or training plans and methodologies, the work of the human engineer had direct and significant impact on these issues. The difference between the knowledge, skills, and abilities required to be a system user and operator and the knowledge, skills, and abilities possessed by prospective users and operators will determine the training and selection requirements. As the designer of all parts of the system with which the human operators and users interact, the human engineer has a direct influence on the training requirements. Additionally, human interfaces can be designed to provide for either ease-of-use or ease-of-learning. It is rare to be able to maximize both of these qualities, and their relative importance will influence the design of tasks, interfaces, and teams, all of which will in turn influence required training.

As the human tasks and interfaces are developed, the human engineer must be aware of constraints on training and selection. The knowledge, skills, and abilities expected to be available in prospective users and operators must be agreed upon by the human engineer and systems engineer. Requirements and constraints for the life cycle support of the system must be available to the human engineer to ensure that the training and selection requirements are compatible. Requirements such as those for on-the-job training or embedded training must be stated early to reduce the likelihood of design changes to meet these requirements at a later date.

IEEE 1220-1998: 6.1.2 – Define project and enterprise constraints
6.1.3 – Define external constraints
6.5.4 – Assess life cycle quality factors
SE OSDs: SE130 – Identify Constraints and Analyze Operational Requirements

7.3 Tradeoff of Concepts and Designs

Once estimates of subsystem or component performance are available, different design alternatives can be traded off to determine the best available option. If multiple alternatives meet the system's functional and performance requirements, then those alternatives should be compared to select the optimal design. Typically, different options will have different strengths and weaknesses, so choosing an option that is strong in one area may decrease performance in other areas. For this reason it is important to have already determined the relative importance of the different design criteria to be used. Even if a formal trade study approach is not employed, the definition of the design criteria will help to justify the selections and make it easier to deal with subsequent changes to system design.

Performing tradeoffs at the component level is typically a simpler task than doing so at higher levels of system design. The interactions between components and subsystems can increase drastically as higher-level designs are considered. Trade studies are typically easier to perform for designers who are only responsible for a single subsystem or feature of the system. The same group of people may have performed all of the design work, and common models and metrics are often used. Due to different models, techniques, and criteria used within different disciplines, trade studies can be more difficult for the systems engineer to perform. The systems engineer has to integrate the different models, data, and criteria that have been employed by the different disciplines or design teams.

In some cases, a tradeoff may involve the decision of whether or not to redesign portions of the system or the degree of redesign required. In such situations, the availability of resources such as time, money, and personnel become as important as technical feasibility. The systems engineers and designers within different disciplines, such as human engineering, must operate from the same set of resource assumptions in making these decisions. In proposing a design change, the human engineer needs to not simply state that there is a problem with the current design, but a potential alternative to the current design should also be provided. This alternative should be in line with the available resources and the selected design criteria for the project as a whole. Simply because the human engineer has the time and resources to make a design change does not mean that the other designers required to implement the change have the available resources.

IEEE 1220-1998: 6.7.5 – Define trade-off analysis scope
SE OSDs: SE320 – Evaluate and Select Preferred Architecture

8 User and Requirements Review

Throughout the system development process, the system design must be reviewed with respect to both its requirements and the operational need. The system design must be compared to all requirements, not simply the top-level system requirements. Designers or verifiers within individual design disciplines must carry out some of this verification process. The conformance or nonconformance of the system design to its requirements must be reported to the systems engineer, who will determine the appropriate course of action based on variables such as system performance and available resources.

8.1 Comparison to Human Engineering Requirements

As system designs are generated from requirements, those designs must then be verified to ensure that the requirements are satisfied. This verification is likely to be at least partially included in the responsibilities of designers in different disciplines. The originators of the requirements, the individuals who created the design, and the design verifiers may be the same people or each may be different.

It is highly probable that the human engineer will need to assess and verify designs generated by others. The specific human engineering requirements, such as design requirements and human performance requirements, must be used to evaluate the designs. A large amount of the verification process will typically be spent on task or job designs or equipment design specific to human engineering. Other designs, however, will have to be reviewed for compatibility with human engineering requirements. Verification may be performed through a variety of different means, ranging from inspection to modeling and simulation to user-in-the-loop testing.

IEEE 1220-1998: 6.6.2 – Conduct verification evaluation
SE OSDs: SE310 – Synthesize Multiple Physical Architectures

8.2 User Review

Verification that the design of a system conforms to requirements is important, but the system design must also be validated. The system needs to conform to the needs of the users, operators, or purchasers, and precise conformance to written requirements does not always provide such assurance. Reviewing potential designs with intended users and operators through means such as storyboards, simulations, and mock-ups can provide early and rapid validation feedback. Full validation that the system meets the operational need may not occur until the system is operational and fielded.

One of the major roles of the human engineer is to determine the requirements and needs of the intended operators and users. Although reviewers such as representative users and operators or subject matter experts may be able to provide some feedback or requirements and functional descriptions, more effective feedback can be generated from the review of proposed physical designs. The human engineer typically has responsibility for human-in-the-loop testing and user reviews. Through system use scenarios and static or dynamic models of system operation, the human engineer can elicit feedback that may be used for changes to designs or requirements. It is frequently useful for the systems engineers and other designers to participate in or observe user testing. Not all feedback will be relevant or valid. Changes to system design or requirements should be based on an objective analysis of information, not on the subjective preferences or opinions of reviewers. The human engineer will need to evaluate the feedback to determine what changes may be considered, and an initial estimate of the impact of those changes on other portions of the system should be made. This information will need to be passed to the systems engineers or other designers.

IEEE 1220-1998: 6.5.11 – Develop models and prototypes
6.6.2 – Conduct verification evaluation
SE OSDs: SE330 – Integrate System Physical Configuration

8.3 Recommendation of Changes to Requirements or Designs

Deficiencies in system design that are revealed through verification or validation must be addressed by some combination of changes to the design and changes to requirements. These changes can frequently have far-reaching effects, leading to time delays and cost overruns. It is the role of the systems engineer to work to balance the required changes with the available resources to meet the design goals. This requires rapid feedback from designers within various disciplines on the impact of changes. The systems engineer must consolidate this feedback and determine the best course of action.

The human engineer should go beyond singling out design deficiencies and should work to present alternative designs or requirements. In some cases, it may be found that the operators simply cannot meet the specified human performance requirements or that unsatisfactory workload levels exist. This will necessitate either a change to the requirements or an addition to the design to provide additional support. Proposed designs may conflict with requirements that have been specified by the human engineer. In some instances, other designers or the systems engineer may want to delete or ignore some requirements related to human engineering. The human engineer must know which human engineering requirements can be traded away to efficiently meet overall system requirements and which requirements cannot be sacrificed. The human engineer should not blindly hold to requirements to optimize human performance when the overall performance of the system will suffer.

Once deviations from requirements are noticed, the human engineer should estimate the impact on overall system performance and begin to develop a resolution to the problem. The proposed changes, however, must have content and be in a format useful to the systems engineer, who will be responsible for resolving conflicts. The format of the suggestions should be identical or at least similar to that of suggestions provided by designers in other disciplines. The content of the suggestions needs to be sufficient to provide the systems engineer with a description of the changes, the rationale for the changes, and anticipated impacts on the remainder of the system.

IEEE 1220-1998: 6.7.1 – Assess requirement conflicts
6.7.3 – Assess design alternatives
SE OSDs: SE330 – Integrate System Physical Configuration

Appendix A: Interactions Sorted by IEEE 1220-1998

IEEE 1220-1998 Paragraph	Systems Engineering and Human Engineering Interaction Paragraph
6.1.2 Define project and enterprise constraints	1.1 Selection of Comparison Systems
	2.1 Human Engineering Constraints
	6.3 Development of Interface and Team Concepts or Designs
	7.2 Training Concept Evaluation
6.1.3 Define external constraints	1.1 Selection of Comparison Systems
	2.1 Human Engineering Constraints
	6.2 Selection of Human Interface and Team Guidelines
	6.3 Development of Interface and Team Concepts or Designs
	7.2 Training Concept Evaluation
6.1.4 Define operational scenarios	1.2 System Use Scenarios
6.1.7 Define interfaces	6.1 Points of Human Interface
	6.3 Development of Interface and Team Concepts or Designs
6.1.8 Define utilization environments	1.3 User Environment Characteristics and Effects
6.1.11 Define performance requirements	2.2 Human Performance Requirements and Human Engineering Design Requirements
6.1.14 Define design characteristics	2.2 Human Performance Requirements and Human Engineering Design Requirements
6.3.2 Functional decomposition	3.1 Functional Decomposition
6.3.3 Establish functional architecture	3.2 Review of Functional Architecture
6.5.1 Group and allocate functions	4.2 Early Identification of Mandatory Allocations
	4.3 Development and Approval of Function Allocation Recommendations
6.5.2 Identify design solution alternatives	5.1 Development of the Task List
	5.2 Identification of Task Characteristics, Interactions, and Sequences
	5.3 Selection of Modeling Tools and Techniques
6.5.4 Assess life cycle quality factors	7.2 Training Concept Evaluation
6.5.5 Assess technology requirements	4.1 Consideration of Human Engineering Technologies
6.5.7 Define physical interfaces	6.1 Points of Human Interface
	6.3 Development of Interface and Team Concepts or Designs

Interactions Sorted by IEEE 1220-1998 - continued

IEEE 1220-1998 Paragraph	Systems Engineering and Human Engineering Interaction Paragraph
6.5.11 Develop models and prototypes	5.3 Selection of Modeling Tools and Techniques
	7.1 Individual and Team Workload and Performance Estimation
	8.2 User Review
6.5.15 Finalize design	5.4 Task and Function Audit
	7.1 Individual and Team Workload and Performance Estimation
6.6.2 Conduct verification evaluation	8.1 Comparison to Human Engineering Requirements
	8.2 User Review
6.7.1 Assess requirement conflicts	8.3 Recommendation of Changes to Requirements or Designs
6.7.3 Assess design alternatives	8.3 Recommendation of Changes to Requirements or Designs
6.7.5 Define trade-off analysis scope	7.3 Tradeoff of Concepts and Designs

Appendix B: Interactions Sorted by Systems Engineering OSDs

Systems Engineering OSD	Systems Engineering and Human Engineering Interaction Paragraph
SE110 – Define and Assess Operational Environment	1.1 Selection of Comparison Systems
	1.2 System Use Scenarios
	1.3 User Environment Characteristics and Effects
SE130 – Identify Constraints and Analyze Operational Requirements	2.1 Human Engineering Constraints
	2.2 Human Performance Requirements and Human Engineering Design Requirements
	6.2 Selection of Human Interface and Team Guidelines
	6.3 Development of Interface and Team Concepts or Designs
	7.2 Training Concept Evaluation
SE140 – Identify Functional and Performance Requirements	2.2 Human Performance Requirements and Human Engineering Design Requirements
SE210 – Functional Definition	3.1 Functional Decomposition
	3.2 Review of Functional Architecture
	6.1 Points of Human Interface
SE310 – Synthesize Multiple Physical Architectures	4.1 Consideration of Human Engineering Technologies
	4.2 Early Identification of Mandatory Allocations
	4.3 Development and Approval of Function Allocation Recommendations
	8.1 Comparison to Human Engineering Requirements
SE320 – Evaluate and Select Preferred Architecture	5.1 Development of the Task List
	5.2 Identification of Task Characteristics, Interactions, and Sequences
	5.3 Selection of Modeling Tools and Techniques
	5.4 Task and Function Audit
	6.1 Points of Human Interface
	7.1 Individual and Team Workload and Performance Estimation
	7.3 Tradeoff of Concepts and Designs
SE330 – Integrate System Physical Configuration	5.3 Selection of Modeling Tools and Techniques
	8.2 User Review
	8.3 Recommendation of Changes to Requirements or Designs