

Modeling Human Workload in Unmanned Aerial Systems:

J. J. Moore, TJ Gledhill, E. Mercer and M. A. Goodrich
Computer Science Department
Brigham Young University
Provo, UT

Abstract—Ironically, unmanned aerial systems (UASs) often require multiple human operators fulfilling diverse roles including mission manager, video analyst, unmanned aerial vehicle operator, etc. Although some dispute the utility of the goal [Murphy], it is a long-standing goal of many UAS designers to decrease the number of humans involved in a mission. This paper presents work toward understanding how workload is distributed among humans and across time, and therefore presents the groundwork for understanding whether it is possible to decrease the number of humans involved and, if it is possible, for what types of missions. The approach is to formally model the actors in a UAS as finite state machines, and then distill a graphical model of the interactions among the different team actors. This graphical model is augmented with a set of workload metrics, derived from a review of the relevant literature, and encoded in Java. The Java PathFinder model-checker is then used to create temporal workload profiles for a pair of scenarios, and the results are checked for consistency with known features of the workload profiles.

Index Terms—model-checking, human machine interfaces, Java PathFinder, unmanned aerial systems, wilderness search and rescue, workload measurement

I. INTRODUCTION

Unmanned aerial systems (UASs), ranging from large military-style Predators to small civilian-use hovercraft, usually require more than one human to operate. This is perhaps ironic, but when a UAS is part of a mission that requires more than moving from point A to point B, there are many different tasks that must be performed including operating the UAS, managing a payload (i.e., camera), managing mission objectives, etc. Some even argue that this is desirable because different aspects of a mission are handled by humans trained for those aspects [?], but since human resources are so expensive many argue that it is desirable to reduce the number of humans involved.

However, the question of *how* to reduce the number of humans while maintaining a high level of robustness is an open question. Some progress has been made by including increasing levels of autonomy including automatic path-planning [], automated target recognition [] etc., and careful human factors analyses have been performed to understand how these technologies impact workload, but with results that are often subtle and sometimes counterintuitive [].

We argue that one reason for the limitations of prior work is that measures of workload is that the level of resolution

for measuring workload is too low. For example, although the NASA TLX dimensions include various contributing factors in workload (e.g., physical effort and mental effort), the temporal distribution of workload tends to be “chunked” across a period of time. Secondary task measures can provide a more detailed albeit indirect breakdown of available cognitive resources as a function of time, but without explanatory power for what in the task causes workload peaks and abatement. Cognitive workload measures, including those that derive from Wickens’ multiple resource theory, provide useful information about the causes of workload spikes, but these measures have not been widely adopted partly because of the difficulty in applying them to new systems. Finally, measures derived from cognitive models such as ACT-R are providing more low-level descriptions of workload including a temporal history, but these approaches do not provide quantitative predictions for the resiliency of the system to deviations in the expected task evolution.

This paper presents a model of four human roles for a UAS-enabled wilderness search and rescue task, and is based on prior work on designing systems through field work and cognitive task analyses [?]. We identify seven *actors* in the team: the UAV, the operator and the operator’s GUI, the video analyst and the analyst’s GUI, the mission manager, and a role we call the parent search that serves to connect the UAS technical search team to the other components of the search enterprise. In the next section, we present the formal model of each of these actors using finite state machines, and then discuss how the connections between these state machines defines what we call a *Directed Team Graph* (DiTG) that describes who communicates with whom and under what conditions.

We then identify a suite of possible workload measures based on a review of the literature, and augment the model to be able to encode specific metrics based on a subset of these measures. Using the Java PathFinder model checker, we then create temporal profiles for each of the workload metrics and check consistency of the temporal profiles by associated workload peaks and abatements with likely causes.

II. ACTOR MODELS

COPY FROM SMC PAPER, REFERENCING THAT PAPER AND TELLING THE READER WE COPIED IT FROM THERE. NEED TO EXPLAIN WHAT THE MATH MEANS, IDENTIFYING

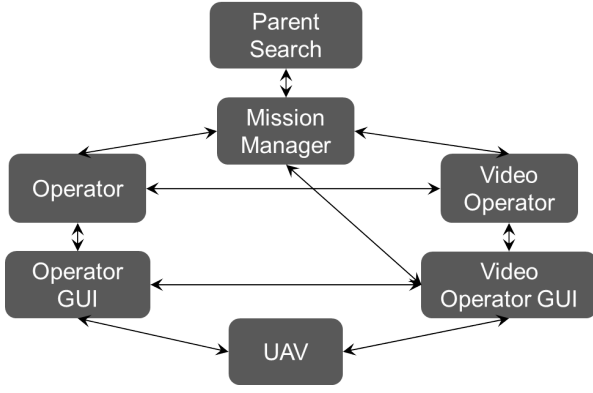


Fig. 1. High Level DiTG

THAT WE ARE USING FSMS AND TALKING ABOUT HOW WE DENOTE THE DIFFERENT ELEMENTS OF THE FSM AND WHAT THEY MEAN IN THE JAVA MODEL.

III. DIRECTED TEAM GRAPH (DiTG)

In order to measure human workload within the context of our model [1] we have defined the following set of core components which allows us to correlate the activity within our model to human workload. We call this framework the directed team graph (DiTG).

A. Conceptual Model

In previous work we represented WiSAR as a collection of directed role graphs (DiRGs)[1]. See figure 1. As is common when modeling human-automation interaction we have decided to model the DiRGs using Mealy state machines which we refer to as Actors [2]. The DiTG formalizes this collection by defining the communication mediums which unite the said Actors. Using multiple resource theory[3] as a guide we can explicitly define the channels over which this communication occurs. See figure 2. The Actor state transitions then use these channels to broadcast and receive inter-Actor communication.

We hope to gain insight into decreasing the system workload, and possibly combining roles, by establishing metrics associated with the system model and model simulation. These metrics can then be used to determine if changes to the model represent a decrease in operator workload.

Formally, the framework is the following mathematical structures:

$$DiTG = (A, \Phi, \forall a_i \in A \exists \Phi_i \subset \Phi) \quad (1)$$

$$Actor = (S, s_0, s_{current}, \Omega_A, \Sigma_A \subset \Phi, \Lambda_A \subset \Phi) \quad (2)$$

$$State = (T_{enabled}, T_{disabled}) : T_{enabled} \cap T_{disabled} = \emptyset \quad (3)$$

$$Transition = (\Omega_{input} \subset \Omega_A, \Sigma_{input} \subset \Sigma_A, \Omega_{value}^{input}, \Sigma_{value}^{input}, \Omega_{output} \subset \Omega_A, \Lambda_{output} \subset \Lambda_A, \Omega_{value}^{output}, \Lambda_{value}^{output}, duration) \quad (4)$$

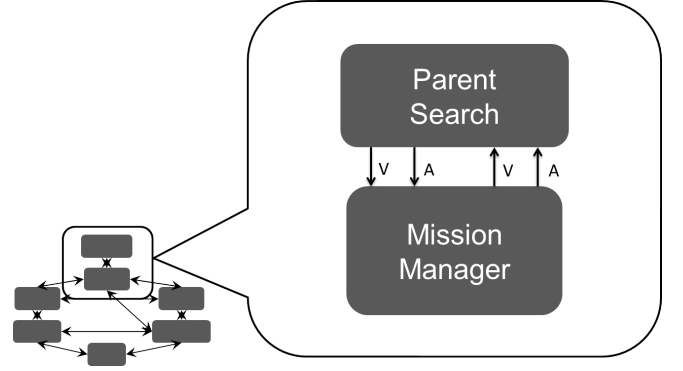


Fig. 2. Detail view of DiTG: V is a Visual channel and A is an Audio channel

$$Channel(\phi) = (type \in (visual, audio), value \in (null, *), a_i^{source}, a_j^{target}) : i \neq j \quad (5)$$

$$DeclarativeMemory(\omega) = value \in (null, *) \quad (6)$$

where A is a set of actors, S a set of states, T a set of transitions, Φ a set of channels, Ω a set of declarative memory, Σ a set of input channels, and Λ a set of output channels.

B. Framework Components

1) *Actors*: Actors represent the agents within the system, while an Actor may be any type of agent for the context of workload we assume that an Actor is human. An Actor is made up of an initial state, a current state, the set of all possible states, declarative memory, input channels and output channels. They are represented within the model as finite state machines.

2) *States*: States contain two sets of Actor transitions, enabled and disabled, which represent all possible transitions the Actor can follow while in that state. While states typically represent the performance of a set of tasks they can also represent such things as emotion, fatigue, and neglect which is expressed in the transitions. In this way an Actor's current state represents its current decision making paradigm.

3) *Transitions*: A transition is composed of a set of required declarative memory and channel values, a set of declarative memory and channel output values, an end state, and a duration. A transition is considered enabled when all of its input requirements are met. The duration represents the relative difficulty of the task(s) associated with the transition. We rationalize this by assuming that all tasks are performed at a constant rate, thus more difficult tasks take longer. It should be noted that our initial models also included transition priority and probability, however, we are ignoring these attributes to simplify our first order workload metrics.

4) *Declarative Memory*: This memory represents internal facts stored by an Actor and used in decision making. This memory takes the form of internal variables within an Actor. Transitions can look for specific values on these variables to determine if they are enabled.

5) *Channels*: Channels represent physical communication mediums that exist between Actors. Each channel has a type (audio or visual), a buffer, a source, and a target. The channel

type is associated with the modality dimension of multiple resource theory while the source and target represent the stages dimension[3]. The source being the response and the target being perception/cognition. We assume that all channels have a constant bandwidth, the longer a channel is in use the more data being sent across the channel.

6) *How it works:* We represent a system as a DiTG, a collection of Actors connected to one another by a set of channels. Whenever the state of the system changes an Actor will petition, from its current state, a list of enabled transitions thus defining what decisions can be made. The Actor may then activate one of these transitions. Transitions have two main states, active and fired. When chosen a transition is made active, after the specified duration the transition fires. When a transition becomes active it creates temporary output values for declarative memory and channels. These temporary values are then applied to the actual declarative memory and channel values once the transition fires.

C. Actor vs Tasks

This architecture focuses on the Actor itself and less on the tasks performed by the Actor. The difference being that Actor states can account for conditions on the Actor which affect performance and increase workload which cannot be seen simply by taking a set of tasks and examining their combined resource requirements. For our model we never explicitly define a single task. Instead we define Actors, States, and Transitions. Each transition defines its own perceptual, cognitive, response, and declarative resources[4] which are not restricted to performing a single task. In this way an Actor's state determines what task(s) are being performed, achieving multi-tasking without explicitly defining tasks. The value of this distinction goes much higher. States can also represent different physical, emotional, or other conditions through adding/removing states and through changes to state transitions. One example is to increase transition durations to represent fatigue. Comparing workload profiles for multiple scenarios using multiple similar models may give insight into what scenarios create higher user workload.

D. Model Creation

To simplify the modeling process and ensure rigorous model creation we developed a transition language, similar to a Kripke structure, which allows models to be expressed as a list of Actor transitions. A parser then automatically generates the classes required to run the model simulation. The transition language uses the following structure.

$$(s_{current}, [\phi_{input} = value, \dots], [\omega_{input} = value, \dots], duration) \times (s_{next}, [\phi_{output} = value, \dots], [\omega_{output} = value, \dots]) \quad (7)$$

IV. WORKLOAD CATEGORIES

Workload categories. 4 Categories In previous research four areas of workload have been well defined. They are Team, Algorithmic, Temporal and Cognitive.

A. Team

MIKE WILL WRITE THIS SUBSECTION AND REVISE THE OTHER SUBSECTIONS IN THE SECTION.

Team Workload is the workload that results from the necessity of working with multiple agents in order to accomplish a task. It can be divided into two sections: group memory and communication. Group memory is

B. Algorithmic

Algorithmic Workload is the result of the difficulty of a task from start to completion. There are three phases these categories can be broken down into: perceive, think and act. During the perception phase the actor takes in all possible inputs and comes up with the list of options available given those inputs. The more options an actor has the greater the workload. Other factors that contribute to the workload of the decision making process are the difficulty of the intended course of action as well as the length of time that said course of action will take.

C. Temporal

Temporal workload deals with the stress of prioritized, infrequent and repetitious tasks. There are three main aspects that contribute to temporal workload. First are the next state constraints. These consist of both timing deadlines as well as the ordering of when tasks are addressed. When a task is constrained by the time it needs to be completed by or the need to complete other tasks in order to respond to the given task it gives the individual an added sense of pressure during the completion. The second category is operational tempo. Operational tempo represents how frequently new tasks arrive. Low and High tempo both cause issues with workload because they frequently result in insufficient time to complete all the given tasks or tasks are forgotten. This can also combine with the concept of fanout, or repetitive tasks. Repetitive tasks contribute to the workload due

D. Cognitive

Cognitive workload describes the difficulties associated with confusing neural signals, this consists of both different channels as well as internal issues such as memory. Parallel Sensing represent the difficulties associated with having multiple channels go high simultaneously. An example of this would be someone's name being called while watching a silent movie. Both the auditory and visual channels are active but the individual is not overwhelmed. The other aspect to Cognitive workload is the necessity of sequential decision making. Conflicts arise between multiple channels each of those channels require decisions to be made based off the input. This results in spikes in the workload level.

V. METRIC CLASSES

Conversion of the Workload theory into quantifiable measurements necessitated the creation of workload metric classes. In the interest of finding areas to consolidate actors we left off measuring team workload. Given that cognitive workload

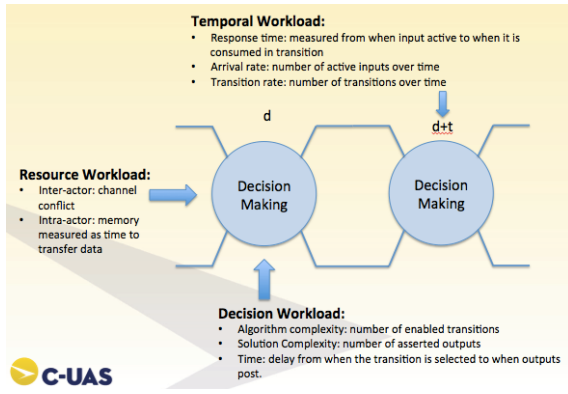


Fig. 3. Workload in the model

describes the difficulties presented by managing resources such as memory and inputs we named that metric class resource. Algorithmic workload analyzes the difficulties presented by decisions giving rise to the decision metrics.

A. Resource Metrics

The model in use lent itself to two distinct ways in to measure Resource Workload. There are a set number of channels connected to each actor and a set amount of data an actor can retain in its internal memory at a time. Counting the memory accesses and channel inputs viewed in each actor state gives a quantifiable metric for the Research Workload. Temporal and the model

B. Temporal Metrics

There are three different ways the model can measure the temporal workload. The number of transitions that occur over a set period of time gives an accurate reflection of the tempo of a given task. The rate at which inputs activate gives a measurement for the arrival rate of new data while the response time can be calculated by measuring the time it takes for an input to be acted upon after it becomes active.

C. Decision Metrics

Decision Workload can be broken down into the timing, the complexity of the algorithm, and the complexity of the solution. Timing is calculated by measuring the time when a transition is chosen till the outputs are posted, this gives us a sense of the duration it takes to complete a given task. The algorithm complexity is calculated by counting how many transitions are possible at a given time while the solution complexity measures the number of outputs that are asserted.

VI. PREDICTING WORKLOAD

In the interest of consolidating operators it is critical to find an accurate measurement that detects situations that exceed the capacity of a given human. One way to detect this is by building a map of each actor's workload as a function of time. JPF is excellent in this regard as it explores all possible paths the model can take and returns the ones that violate the model's criteria. By augmenting our model with the metrics described above we can identify all possible areas of high workload.

Once these critical sections have been identified we will either be able to minimize them by increasing the autonomy of the system or by rearranging task protocols to balance sections of high workload with other lower workload areas.

There are three techniques that we need to undergo in the validation process of our workload measurements. The first is consistency. As we analyze the maps produced in JPF we will need to verify that deviations in the workload are predictable results. For example there should be no spikes in workload when all the actors are at rest. Once we've verified that the workload measurements are consistent with our understanding we will execute a sensitivity study. During this study we will instigate random permutations into the model and verify that the workload mutations correlate with those permutations. This study will verify that our understanding of the workload metrics implemented is accurate. Finally we will need to initiate a human study to verify that our metrics do in fact correlate with workload in humans. In this study we will use situations that caused spikes, low points, and high points of workload. By running our subjects through these same situations and receiving their feedback we will be able to verify that our metrics do in fact measure workload.

VII. RESULTS

empty

VIII. RELATED WORK

This work is an extension of previous work which focused on modeling human machine systems, specifically WiSAR. This work extends this model to incorporate the measurement of workload. [1]

Multiple resource theory plays a key role in how we are measuring workload. [3] The multiple resource model defines four categorical dimensions that account for variations in human task performance. A task can be represented as a vector of these dimensions. Tasks interfere when they share resource dimensions. Using these vectors Wickens defined a basic workload measure consisting of the task difficulty (0,1,2) and the number of shared dimensions. Using this metric it is possible to predict task interference by looking at tasks which use the same resource dimensions. Our model differs in that we do not explicitly define tasks, instead we use Actor state transitions which may imply any number of concurrent tasks. The transition then informs us of which resources are being used and for how long. To simplify our implementation we are currently only looking at two dimensions, Stages and Modalities. We can use the same metric to predict if an Actors transition will cause task interference.

Threaded cognition theory states that humans can perform multiple concurrent tasks that do not require executive processes. By making a broad list of resource assumptions about humans, threaded cognition is able to detect the resource conflicts of multiple concurrent tasks. Additionally threaded cognition is able to model task learning. Our model differs from Threaded cognition theory in a few key areas, our model does not allow learning which decreases workload through skill

mastery. Also, it does not distinguish between perceptual and motor resources, instead perception and action both use the same resource. In almost all other aspects our model behaves in a similar fashion. [4]

Other similar work has attempted to predict the number of UAVs an operator can control, otherwise known as fan-out. [5] This work used queuing theory to model how a human responds in a time sensitive multi-task environment. Queuing theory is helpful in determining the temporal effects of task performance by measuring the difference between when a task was received and when it was executed. At the highest level our model uses Queuing theory, our Actors can only perform a single transition at a time. We differ, however, in that a transition may represent multiple concurrent tasks.

Act-R is a cognitive architecture which attempts to model human cognition and has been successful in human-computer interaction applications. [6] The framework for this architecture consists of modules, buffers, and a pattern matcher which in many ways are very similar to our own framework. The major difference being the cognitive detail available with ACT-R such as memory access time, task learning, and motor vs perceptual resource differences. [7, 8]

IX. CONCLUSION AND FUTURE WORK

For our next steps we have several basic plans. We will also be adding an actor to handle sense and avoidance operations. In addition to that it is necessary to implement a way to evaluate the cost of errors in decision making. There will be certain tasks that cannot be automated because the cost of failure requires a human's undivided attention. First we will be undergoing a sensitivity study, followed by a field test. One we've verified that our system analyzes workload correctly We will branch our research into two separate directions. The first area will consist of generated an optimized GUI for the WiSAR system, hopefully to allow a single operator to take full control of the WiSAR system. While the second branch will be formulated a generalized model that will have application across the board for UAV systems.

ACKNOWLEDGMENT

The authors would like to thank Neha Rungta of NASA Ames Intelligent Systems Division for her help with JPF and Brahms. The authors would also like to thank the NSF IUCRC Center for Unmanned Aerial Systems, and the participating industries and labs, for funding the work. Further thanks go to Jared Moore and Robert Ivie for their help coding the model and editing this paper.

REFERENCES

- [1] *Modeling UASs for Role Fusion and Human Machine Interface Optimization*, ser. SMC'13, 2013.
- [2] *Using Formal Verification to Evaluate Human-Automation Interaction: A Review*, 2013.
- [3] C. D. Wickens, "Multiple resources and performance prediction," *Theoretical issues in ergonomics science*, vol. 3, no. 2, pp. 159–177, 2002.
- [4] D. D. Salvucci and N. A. Taatgen, "Threaded cognition: an integrated theory of concurrent multitasking," *Psychological review*, vol. 115, no. 1, p. 101, 2008.
- [5] M. L. Cummings, C. E. Nehme, J. Crandall, and P. Mitchell, "Predicting operator capacity for supervisory control of multiple uavs," in *Innovations in Intelligent Machines-1*. Springer, 2007, pp. 11–37.
- [6] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind," *Psychological review*, vol. 111, no. 4, p. 1036, 2004.
- [7] N. Moray, "Subjective mental workload," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 24, no. 1, pp. 25–40, 1982.
- [8] A. Newell, *Unified theories of cognition*. Harvard University Press, 1994, vol. 187.