In the 2nd International Conference on Applied Human Factors and Ergonomics, July 14-17 2008, Las Vegas, NV

Meeting the Challenge of Cognitive Human Performance Model Interpretability Through Transparency: MIDAS v5.x

Gore, Brian F.

San Jose State University Research Foundation/NASA Ames Research Center / Human-Systems Integration Division / MS 262-12 / Moffett Field, CA 94035-1000 USA
E-mail: Brian.F.Gore@nasa.gov

Hooey, Becky L.

San Jose State University Research Foundation/NASA Ames Research Center / Human-Systems Integration Division / MS 262-4 / Moffett Field, CA 94035-1000 USA
E-mail: Becky.L.Hooey@nasa.gov

Foyle, David C.

NASA Ames Research Center / Human-Systems Integration Division / MS 262-4 / Moffett Field, CA 94035-1000 USA
E-mail: David.C.Foyle@nasa.gov

Scott-Nash, Shelly

Alion Science and Technology/ Micro Analysis and Design Operation / Boulder, CO 80301 USA
E-mail: sscott-nash@alionscience.com

ABSTRACT Transparency in integrated human performance models (HPMs) is needed to support model verification, validation, and credibility. However, model transparency can be difficult to attain because of the complex interactions that can exist among the cognitive, physical, environment and crewstation models, and because the cognitive models embedded within integrated HPMs produce behaviors that are not directly observable. This paper will illustrate several techniques adopted by the Man-machine Integration Design and Analysis System (MIDAS) to increase three forms of transparency: input transparency, model architecture transparency, and output transparency.

Keywords

Human performance model, physical and cognitive integrated model, Man-machine Integration Design and Analysis System (MIDAS), transparency, interpretability

INTRODUCTION

Modeling human cognition, and understanding the manner that humans use information, is becoming increasingly important as system designers develop automation to support human operators. Tasks that were traditionally manual control and physical in nature are being replaced with tasks that are cognitive in nature. This is exemplified in the aviation community where automation is being adopted to increase efficiency and safety (NextGen, 2007). It is also becoming more important in surface transportation with automation such as adaptive cruise control, autonomous cruise control and vehicle guidance replacing manual control requirements with an increased need to monitor the automobile’s performance (Sheridan, 1992; Seppelt & Lee, 2007). Furthermore, nuclear power plant design, medical system design and operation, unmanned aerial vehicles (UAVs) and other telerobotic operations and manufacturing systems are all increasingly placing the human into similar supervisory roles (Sheridan and Ferrell, 1974; Miller, 2000; Zhai & Milgram, 1991; Sheridan, 1992; Boring et al., 2006). In these environments, it is important to model both the physical human as well as human cognition.

Human performance models (HPMs) can take many forms from the purely cognitive models built from empirical research and theories of human processes (e.g., attention, perception, decision making, response times, and response characteristics), to digital human models (DHMs), or physical models of human anthropometry, biomechanics, posture, movement, bones and anatomy (Gore, 2008). Incorrectly modeling any of these performance factors, or ignoring one in favor of the other, may lead to incorrect predictions. Cognitive and physical HPMs can be combined to produce an integrated HPM that simulates human responses (Gore, 2008) and predict how humans interact with advanced technologies (Gore & Smith, 2006).

Integrated HPMs combine a number of individual process models of operator performance into a coordinated representation of interacting micromodels of human perceptual, cognitive and human motor system representations. These micromodels feed-forward and feedback to each other within the HPM's environmental context to produce predictions of operator performance[1]. In addition, some of the more comprehensive integrated models incorporate computer-aided design / engineering (CAD/E) renditions of the environment, the crewstation, and the human form. These DHMs can feed physical constraints to cognitive process models that in turn feed information to the DHM in a closed-loop fashion.

One example of an established integrated HPM is the Man-machine Integration Design and Analysis System (MIDAS)[2]. MIDAS is a 3-D rapid-prototyping human performance modeling and simulation environment that can be used to facilitate the design, visualization, and evaluation of complex human-system concepts. MIDAS aims to reduce design cycle time, support quantitative predictions of human-system effectiveness, and improve the design of crew stations and their associated operating procedures. MIDAS accomplishes integrated behavioral modeling by linking a virtual human (a physical anthropometric mannequin model) to a computational cognitive structure representing human capabilities and limitations, and to a series of procedures, and places this virtual human within commercially available CAD/E databases. MIDAS combines continuous-control, discrete-control and critical decision-making models to represent the 'internal models and cognitive function' of the human operator in complex control systems. It involves a critical coupling among humans and machines in a shifting and context-sensitive function. MIDAS' "first principles" approach to modeling human performance is based on computational models of the mechanisms that underlie and cause human behavior within the context of human-system performance (Corker & Smith, 1993; Gore & Corker, 2000). The basic human perceptual and attentional processes, together with working and long-term memory models[3], action selection architectures and physical representations of the human operator and environment models have been validated[4]. The closed-loop nature of the relationship between cognitive models and DHMs results in more representative human-system performance than either approach alone.

Need for Transparent and Interpretable Human Performance Models

Model transparency refers to the ability to comprehend the relationships that exist among the models being used in the simulation, the performance of the models in the simulation, which models are triggering in the model architecture, and whether

[1] For a complete description of the component models within an integrated representation, the reader is directed to (Gore, 2002), Gore & Smith (2006) and Gore (2008).

[2] The MIDAS research program began in the fall of 1984 (at the time termed A^{3}I) and began developing the first fully integrated HPM linking together cognitive and performance models.

[3] MIDAS v5.0.2 (beta) currently includes a three-stage memory model with a working memory, long-term working memory and long-term memory representation.

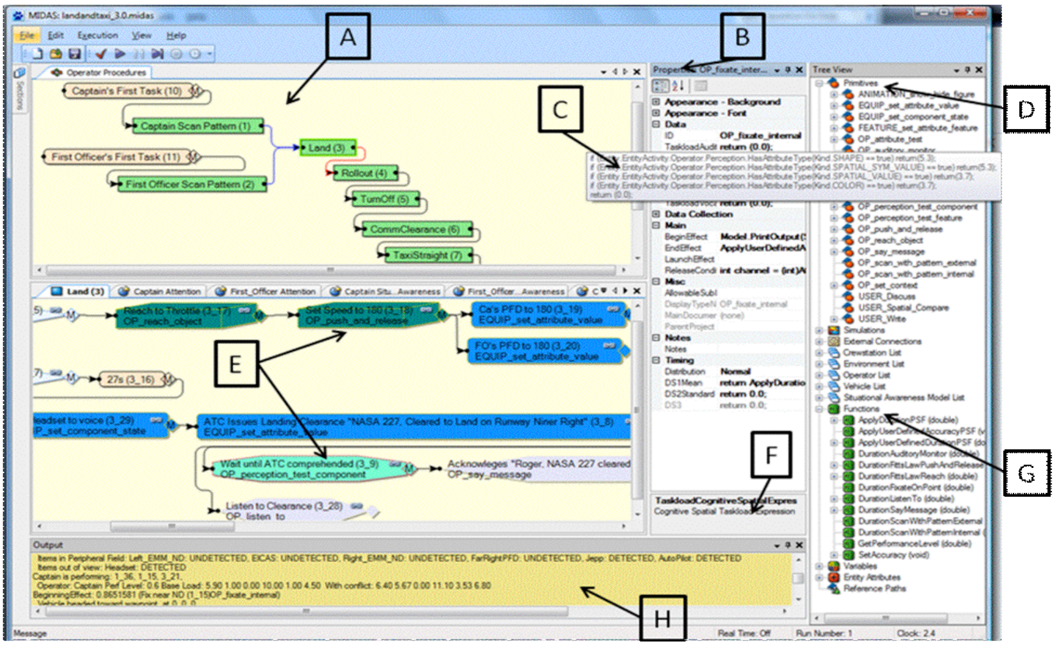
[4] See Gore & Corker (1999) for a listing of the validated models in MIDAS.

the model is behaving as the model developer would expect (Gore, 2008). Gluck and Pew (2005) refer to this as runtime interpretability. Other researchers have referred to this as model traceability, model behavior visibility, model verifiability, model validity and model interpretability (NASA, 1989; Napierky, Young, Harper, 2004; Gluck & Pew, 2005; Hooey & Foyle, 2008).

Combining models in an integrated fashion, as in MIDAS, adds complexity. This complexity highlights the need for improved model transparency via visualization tools that maximize comprehensibility of the models operating within the software. Model transparency is particularly important for models that include representations of human cognition because cognition is an internal process that is not directly observable. In addition to increasing the visibility of the cognitive processes, model transparency is required to support model verification and validation, which, in turn helps establish model credibility. When models are transparent the user has increased confidence that the output from the model is in line with human performance. Conversely, when models are not transparent, the user cannot confidently interpret, nor trust that the model's performance is operating according to his/her expectations. While there is widespread agreement as to the importance of model transparency, there is less consensus regarding what transparency actually means, and how to accomplish it. This paper outlines three classes of model transparency: Input Transparency, Model Architecture Transparency, and Output Transparency. A discussion of the importance of each class of transparency, and how each was accomplished in MIDAS, follows.

INPUT TRANSPARENCY

Input transparency refers to visualization techniques that allow the user to inspect the procedures and data that have been input into the model. In many instances, model inputs consist simply of lines of code, which are difficult to inspect not only by the modelers themselves but also by other users of the model output – especially those that lack modeling and/or programming skills. To address this problem and increase input transparency, MIDAS 5.x was augmented to include the modeled operator's procedures in a task network manner. An example of the task network depiction is provided in Figure 1, location A, which depicts the high-level procedures of a Captain and First Officer conducting a land-and-taxi-to-the-gate procedure.


MIDAS: landandtaxi_3.0.midas
File Edit Execution View Help
A
B
C
D
E
F
G
H
Operator Procedures
Captain's First Task (10)
Captain Scan Pattern (1)
Land (3)
First Officer's First Task (11)
First Officer Scan Pattern (2)
Rollout (4)
TurnOff (5)
CommClearance (6)
TaxiStraight (7)
Land (3)
Captain Attention
First_Officer Attention
Captain Situ...Awareness
First_Officer...Awareness
Reach to Throttle (3_17)
OP_reach_object
Set Speed to 180 (3_18)
OP_push_and_release
Ca's PFD to 180 (3_19)
EQUIP_set_attribute_value
FO's PFD to 180 (3_20)
EQUIP_set_attribute_value
27s (3_16)
Headset to voice (3_29)
OP_set_component_state
ATC Issues Landing Clearance "NASA 227, Cleared to Land on Runway Niner Right" (3_8)
EQUIP_set_attribute_value
Wait until ATC comprehended (3_9)
OP_perception_test_component
Acknowleges "Roger, NASA 227 cleared
OP_say_message
Listen to Clearance (3_28)
OP_listen_to
Properties OP_fixate_inter...
Appearance - Background
Appearance - Font
Data
ID OP_fixate_internal
TaskloadAudi return (0.0);
TaskloadVoc return (0.0);
Data Collection
Main
BeginEffect Model.PrintOutput(
EndEffect ApplyUserDefinedA
LaunchEffect
ReleaseCondi int channel = (int)A
Misc
AllowableSubl
DisplayTypeN OP_fixate_internal
MainDocumer (none)
ParentProject
Notes
Notes
Timing
Distribution Normal
DS1Mean return ApplyDuratio
DS2Standard return 0.0;
DS3 return 0.0;
TaskloadCognitiveSpatialExpres
Cognitive Spatial Taskload Expression
Tree View
Primitives
ANIMATION_show_hide_figure
EQUIP_set_attribute_value
EQUIP_set_component_state
FEATURE_set_attribute_feature
OP_attribute_test
OP_auditory_monitor
OP_perception_test_component
OP_perception_test_feature
OP_push_and_release
OP_reach_object
OP_say_message
OP_scan_with_pattern_external
OP_scan_with_pattern_internal
OP_set_context
USER_Discuss
USER_Spatial_Compare
USER_Write
Simulations
External Connections
Crewstation List
Environment List
Operator List
Vehicle List
Situational Awareness Model List
Functions
ApplyOperatorPSF (double)
ApplyUserDefinedAccuracyPSF
ApplyUserDefinedDurationPSF
DurationAuditoryMonitor (double)
DurationFittsLawPushAndRelease
DurationFittsLawReach (double)
DurationFixateOnPoint (double)
DurationListenTo (double)
DurationSayMessage (double)
DurationScanWithPatternExternal
DurationScanWithPatternInternal
GetPerformanceLevel (double)
SetAccuracy (void)
Variables
Entity Attributes
Reference Paths
Output
Message
Real Time: Off
Run Number: 1
Clock: 2.4

```
if (Entity.EntityActivity.Operator.Perception.HasAttributeType(Kind.SHAPE) == true) return(5.3);
if (Entity.EntityActivity.Operator.Perception.HasAttributeType(Kind.SPATIAL_SYM_VALUE) == true) return(5.3);
if (Entity.EntityActivity.Operator.Perception.HasAttributeType(Kind.SPATIAL_VALUE) == true) return(3.7);
if (Entity.EntityActivity.Operator.Perception.HasAttributeType(Kind.COLOR) == true) return(3.7);
return (0.0);
```

```
  Items in Peripheral Field: Left_EMM_ND: UNDETECTED, EICAS: UNDETECTED, Right_EMM_ND: UNDETECTED, FarRightPFD: UNDETECTED, Jepp: DETECTED, AutoPilot: DETECTED
  Items out of view: Headset: DETECTED
Captain is performing: 1_36, 1_15, 3_21,
  Operator: Captain Perf Level: 0.6 Base Load: 5.90 1.00 0.00 10.00 1.00 4.50  With conflict: 6.40 5.67 0.00 11.10 3.53 6.80
BeginningEffect: 0.8651581 (Fix near ND (1_15)OP_fixate_internal)
Vehicle headed toward waypoint at 0,0,0
```

Figure 1. MIDAS 5.x Windows - Operator Procedures (A), Properties (B), Primitive Taskload (C), Primitives in Tree View (D), Usage of Primitives (E), Notes (F), Functions (G) and Code Output (H).

These high-level procedures can be populated with additional logic through a sequence of nested activity trees. The analyst can see the tasks and task sequence of each operator, and drill down for increased detail and sub-task breakdowns. For example, Figure 2 highlights the auditory and visual activities that the Captain will complete in carrying out the landing task. This network structure shows that the Captain is monitoring auditory information and in parallel carrying out an internal scan pattern using a probabilistic sequence of fixation locations for the landing task. Once the landing task is completed, a sequence of the other tasks is completed depending on the context of the operator (rollout, turnoff, communication, taxiing straight, turning, aircraft at gate).

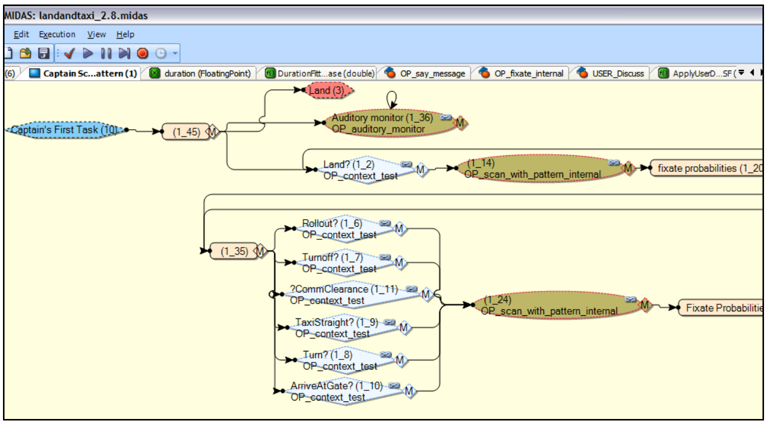


Figure 2. Increased detail of the sub-task breakdown of a MIDAS v5.x model

Another tool by which input transparency is offered to the MIDAS user is the runtime visualization of the anthropometric figure, Jack™[5], carrying out the scheduled tasks and procedures. This form of visualization is considered input transparency because it allows the model analyst to verify that the sequence of behaviors that have been programmed into the model is correctly ordered and that the operator is engaging in the correct behaviors at the correct simulation time. Figure 3 illustrates two modeled operators conducting the series of procedures necessary to land an aircraft. The visual cones illustrated in Figure 3 are examples of one manner in which an internal cognitive process, visual attention, can be made transparent using runtime graphical visualizations. The visual cones in the Jack™ software are graphic renditions of an approximate peripheral and foveal location. The MIDAS behavior model drives this cone with an empirical model of human peripheral and foveal vision response time. This visualization allows the MIDAS analyst to troubleshoot the input procedures to verify the order of the procedures and tasks and the internal and external scans. The benefit of the visualization for the simulation analyst is that it can be used to examine whether the procedure was correctly transferred and used by the model. Often times, it is not possible to make sense of the logic contained within the fast time simulation code without this kind of visualization. This is especially true as models are becoming more complex and greater numbers of coordinating operators are working towards a common system goal.

These input transparency tools allow a model analyst to inspect the model and identify erroneous tasks or task sequence that may contribute to the generation of false or suspect conclusions (Hooey & Foyle, 2008). Without some visualization or trace of the modeled operator's tasks, catching erroneous or omitted tasks in the model would be very unlikely, even by the modeler, given the complexity of the computer code required to implement a complex model (Hooey & Foyle, 2008; see also Gore, 2008; Gluck & Pew, 2005).

[5] ™ Siemens PLM Solutions, Inc

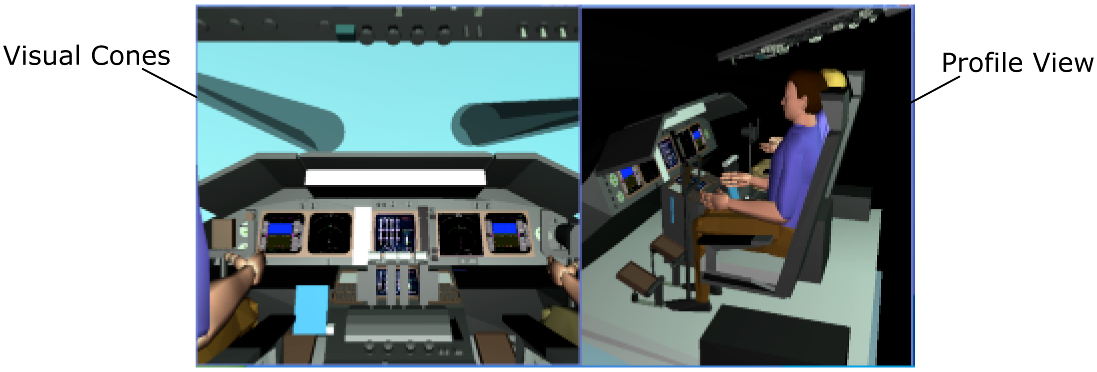


Figure 3. Runtime visualization of physical and anthropometric models carrying out procedures in the Jack™ environment. The left panel illustrates the view from the center of the cockpit and the right panel illustrates the left seat pilot's profile.

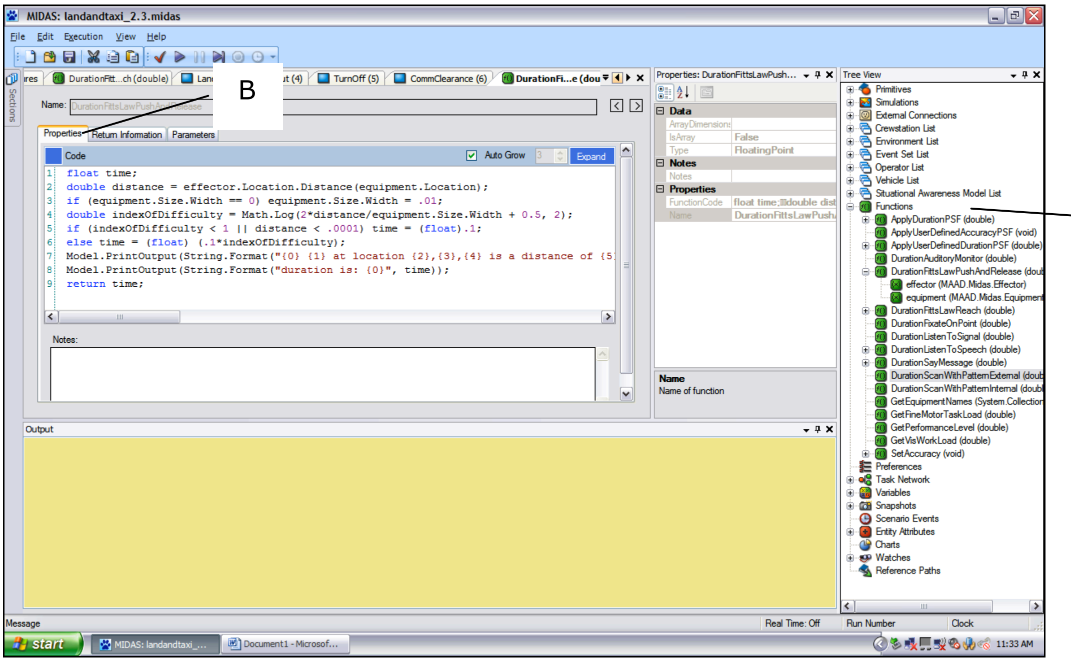
MODEL ARCHITECTURE TRANSPARENCY

Model architecture transparency, arguably the most difficult form to attain, refers to depictions of the underlying submodels, logic, and interactions among submodels. As models increase in complexity, both in terms of the embedded models contained within their architecture and with respect to the environments to which their results are being applied, it is increasingly difficult to interpret the empirical basis behind the algorithms driving the modeling architecture. When models are developed that integrate a number of submodels together, as in MIDAS, the model software should be transparent such that a user can determine which submodels are active at any given time in a scenario (NASA, 1989; Pew, Gluck, & Deutsch, 2005). In an evaluation of two distinct HPM software tools, Gore and Corker (2000) raised this issue as a challenge for integrated HPM workload models. As discussed in Leiden and Best (2008), in some HPMs, individual micromodels can be engaged or disengaged for many reasons. If behaviors become too constrained by some rule, the model developer can turn off this constraint, thereby allowing the model to proceed to completion within a given architecture. Alternatively, the model developer can remove the effect of certain elements of human performance (presumably deemed to be non relevant) by turning elements of a model on or off. For example, to remove the effect of memory on human performance, one might disengage the memory model thereby ensuring perfect memory. As such, it is often difficult to ascertain which micromodels were operating during runtime and which were not, and what the resultant effects on predicted behavior would be. Without this level of insight into the model, and an accurate understanding of the assumptions embedded in the model, results may be overstated. As Hooey and Foyle (2008) point out, this is a significant problem for some integrated modeling software tools because no definite statement can be made as to which performance parameters were operational within the architecture when the modeling software was run.

The difficulty associated with verifying and tracing the performance of the various micromodels contained within early versions of MIDAS motivated the NASA MIDAS team to improve the model architecture transparency in MIDAS 5.x. MIDAS was augmented by adding a tree view of all of the model components including the behavioral primitives (Figure 1, location D). The primitives represent empirically based, human performance models within the MIDAS software. This "primitives" list is used by the model analyst to inspect the available operator models, equipment models, or user-defined models that are available for use in the simulation. The properties/computational logic associated with these behavioral primitives is also illustrated in Figure 1, location B. This properties tab illustrates the logic and relationship among the various model components. The properties tab also possesses the taskload levels that are associated with the component model within MIDAS, the beginning and ending effect, and the if-then release conditions associated with the component model being triggered among other logical conditions.

For example, Figure 1, location D illustrates the operator primitives available to drive the performance of the modeled operator in the simulation and Figure 1, location E demonstrates usage of the primitives in a task sequence. One example from Figure 1 is the operator primitives of "push and release" and "reach object". These behavioral primitives used in sequence drive the operator model to complete a reach followed by a push and release action to a button in the environment. This function animates the anthropometric model, contributes a defined taskload to operator workload and calculates a task time based on the distance to and size of target to which the modeled operator is reaching. A second example, "attribute test" combined with "perception test", returns the value of an attribute along with the operator's current perception level of a component in the crewstation or a feature in the external environment that contains that attribute. This means that the operator will carry out the primitives necessary for the perception model to reach a threshold that will allow perception of a feature or component in the environment to be attained. In addition, the perception workload is driven by the type of attributes displayed or emitted by a component or feature of a model (see Figure 1, location C). The models listed in these primitives comprise the basic models within MIDAS and because of their importance to driving correct human responses, cannot be changed by the user. Users can add new primitive behaviors in the "user-defined" areas of the software, which MIDAS 5.x reports along with the output because these models possess an increased likelihood of affecting other models contained within MIDAS' integrated architecture. These user-defined primitive models or functions are then vetted for compliance with the MIDAS assumptions and if the models/functions are within acceptable limits, the models/functions will be included in future versions of MIDAS. This is important because only with a consistent model architecture can the results be interpreted or compared across tasks and domains.

A host of functions exist within MIDAS 5.x but a subset includes visual and auditory perception functions, physical performance functions, and performance shaping factors such as fatigue, stress, workload levels and performance accuracy. As shown in Figure 4, location A (also shown in Figure 1, location G), functions are presented in a "tree view" tab under the "functions" folder and the algorithms contained within the functions are interpreted by viewing the "properties" window of the specific function of interest (e.g., Fitts's Law – Figure 4, location B). It is also linked to a "Notes" page (not shown) that provides the user with the empirical research results that were used as the basis for developing the MIDAS primitives and algorithms. Fitts's Law is one of the component models within MIDAS. It is called whenever an operator engages in a manual control behavior such as a reach or small hand movement. The modeled operator's performance is a function of the distance that the hand travels and the size of the target he/she is reaching towards (Fitts, 1954). The MIDAS analyst needs to know this function so that he/she can have confidence in the performance of the integrated model. When the analyst right-clicks on the specific function labeled "Fitts's Law", the algorithmic information that makes up the Fitts's Law function in MIDAS is revealed.


MIDAS: landandtaxi_2.3.midas
File Edit Execution View Help
DurationFitt...ch (double)
TurnOff (5)
CommClearance (6)
DurationFi...e (dou
B
Name: DurationFittsLawPushAndRelease
Properties
Return Information
Parameters
Code
Auto Grow
Expand
Notes:
Properties: DurationFittsLawPush...
Data
ArrayDimensions
IsArray False
Type FloatingPoint
Notes
Properties
FunctionCode float time;double dist
Name DurationFittsLawPush
Name
Name of function
Output
Tree View
Primitives
Simulations
External Connections
Crewstation List
Environment List
Event Set List
Operator List
Vehicle List
Situational Awareness Model List
Functions
ApplyDurationPSF (double)
ApplyUserDefinedAccuracyPSF (void)
ApplyUserDefinedDurationPSF (double)
DurationAuditoryMonitor (double)
DurationFittsLawPushAndRelease (doub
effector (MAAD.Midas.Effector)
equipment (MAAD.Midas.Equipment
DurationFittsLawReach (double)
DurationFixateOnPoint (double)
DurationListenToSignal (double)
DurationListenToSpeech (double)
DurationSayMessage (double)
DurationScanWithPatternExternal (doub
DurationScanWithPatternInternal (doubl
GetEquipmentNames (System.Collection
GetFineMotorTaskLoad (double)
GetPerformanceLevel (double)
GetVisWorkLoad (double)
SetAccuracy (void)
Preferences
Task Network
Variables
Snapshots
Scenario Events
Entity Attributes
Charts
Watches
Reference Paths
A
Message
Real Time: Off
Run Number
Clock
start
MIDAS: landandtaxi_...
Document1 - Microsof...
11:33 AM

```
float time;
double distance = effector.Location.Distance(equipment.Location);
if (equipment.Size.Width == 0) equipment.Size.Width = .01;
double indexOfDifficulty = Math.Log(2*distance/equipment.Size.Width + 0.5, 2);
if (indexOfDifficulty < 1 || distance < .0001) time = (float).1;
else time = (float) (.1*indexOfDifficulty);
Model.PrintOutput(String.Format("{0} {1} at location {2},{3},{4} is a distance of {5
Model.PrintOutput(String.Format("duration is: {0}", time));
return time;
```

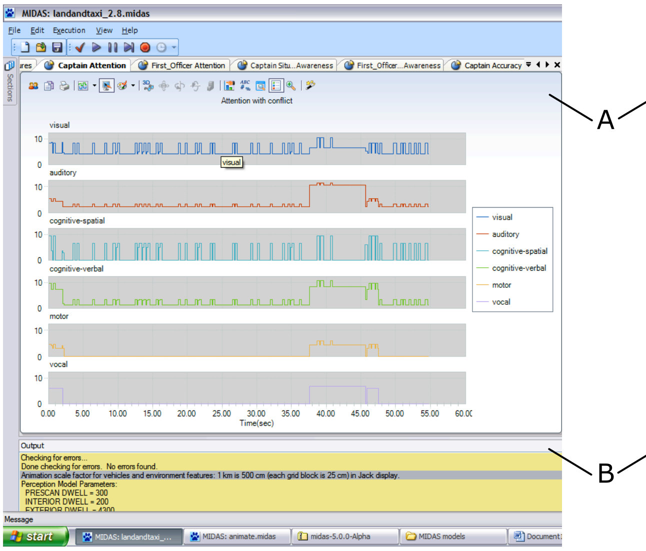
Figure 4: MIDAS 5.x Windows - function list (A) and function properties (B).

OUTPUT TRANSPARENCY

Output transparency refers to methods and techniques that use the model output to provide visibility into the internal operations of the model. In the case of MIDAS, output includes task timing, task order, as well as psychological constructs such as workload, and situation awareness (SA). Investigating the construct output of workload and SA in isolation, without considering the operators tasks, can be a challenge with integrated models because of the interactions among the models contained within the integrated model structure. Runtime visualizations of the anthropometry provide the model analyst with increased confidence that the model's input parameters and tasks are occurring as would be expected but do not provide enough information about the true effect of the system variables' impact on the human. For that, runtime output of workload and SA metrics are provided in MIDAS 5.x. An example of the workload output is presented next.

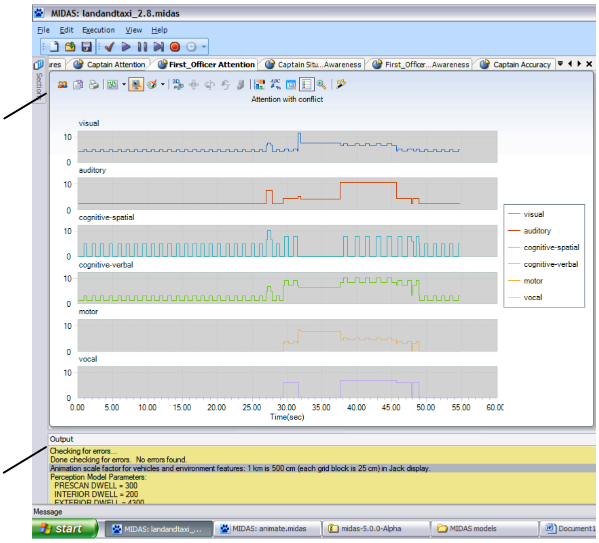
Workload is defined as the perceived relationship between the amount of mental processing capability or resources and the amount required by the task (Hart & Staveland, 1988). As reported in Gore (2008), MIDAS calculates attention demands based on Wickens' (1984) Multiple Resource Principle and incorporates a task-loading index initially created by McCracken and Aldrich (1984) for quantifying attention. This scale was modified subsequently to include a six-channel representation of task load. Combining attention demands along the input (visual, auditory), central cognitive processing (spatial, verbal processing), and output (psychomotor, vocal) resources accomplish a measure of attention demands. In addition, MIDAS degradation functions were developed to model the effects of a stressor on skilled performance through workload exceedances. Wickens (1984) and Hamilton, Bierbaum, and Fulford (1990) indicate that when workload exceeds the theoretical threshold value of 7, the operator sheds tasks resulting in a performance degradation.

Examining workload values in isolation, without consideration of the operators' task sequence, results in a loss of valuable scenario context information. To retain the scenario context, output transparency is offered in MIDAS 5.x by enabling graphical depictions of workload as a function of the simulation timeline. Figures 5 and 6 location A provide insight into the interactive nature of two operators in a simulation by providing a trace of the workload (noted as attention demands) model in a recently completed aviation simulation of two pilots performing the land-and-taxi-to-gate scenario shown above. The two figures illustrate operator workload levels along the 6-channel workload output characterization used by MIDAS. As model verification, it is apparent that the workload (attention) demands possess characteristic profiles depending on the operator's role in the cockpit with workload spikes occurring at different times in the scenario commensurate with each pilot's procedural requirements. Furthermore, the workload trends can be interpreted by examining the ongoing operator tasks in the code output window that denotes the model logic used to generate the output (Figure 1, location H; Figures 5 and 6, location B). The code output window is yet another tool that can be used along with the other visualization tools to verify the operation of the model. When these output measures are used in concert with the procedural sequences of the respective operator, periods of system vulnerability can be identified and procedural redesigns recommended with increased confidence that the model validly produced behaviors.



```
Checking for errors...
Done checking for errors.  No errors found.
Animation scale factor for vehicles and environment features: 1 km is 500 cm (each grid block is 25 cm) in Jack display.
Perception Model Parameters:
 PRESCAN DWELL = 300
 INTERIOR DWELL = 200
 EXTERIOR DWELL = 4300
```

Figure 5. Captain Workload (Attention) output (top) and Code Output (bottom)



```
Checking for errors...
Done checking for errors.  No errors found.
Animation scale factor for vehicles and environment features: 1 km is 500 cm (each grid block is 25 cm) in Jack display.
Perception Model Parameters:
 PRESCAN DWELL = 300
 INTERIOR DWELL = 200
 EXTERIOR DWELL = 4300
```

Figure 6. First Officer Workload (Attention) output (top) and Code Output (bottom)

DISCUSSION

Balancing Model Transparency and Software Flexibility

One of the largest challenges that the MIDAS model development team faced in developing MIDAS v5.x was balancing software transparency and model flexibility while maintaining control of the theoretical underpinnings of the MIDAS software and of the MIDAS philosophy. Allowing the MIDAS users/analysts the ability to enter primitive behaviors, or modify the algorithms through the function calls could compromise the output generated by the generally accepted "MIDAS standard" models, which in turn could compromise the confidence of the software's output. The interactions among the generally accepted "MIDAS standard" models and the user-defined models cannot possess transparency because the interactions are not well known or defined. Resultantly, the MIDAS development team established a number of fundamental models that cannot be edited by the model user/analyst. Those user-defined models that are vetted by NASA's MIDAS team become integrated into subsequent releases of the MIDAS software. The vetting process supports model credibility.

The complexity of human-system interactions is certain to continue as the human is placed into roles with increasingly shared responsibility and authority between the human and automation as exemplified by the next generation airspace, next generation process control room designs, or automated highway systems. These systems are driving minimum thresholds for the human's performance, which are constrained by human cognition. Integrated models, such as MIDAS, are useful tools to explore the degree to which the human operates in the realm of these threshold values, which in turn will contribute to system vulnerabilities. As these complex HPMs are used in advanced human-system designs, the need for model transparency is heightened because of the complexity of the interactions. The current paper has illustrated the strides that have been taken to improve the transparency of MIDAS. MIDAS has addressed model interpretability by attaining multiple levels of transparency (input, model architecture, and model output), which has improved the confidence that the model analysts/users can place on the output of the software.

CONCLUSION

The complexity associated with interactions among the many frequently imperceptible cognitive aspects that guide human behavior will remain a challenge for the field of cognitive modeling for years to come. For the field of human performance modeling to advance, it is critical that a comprehensive understanding of the mechanisms operating in the model is formalized, and that there is sufficient transparency in the model's operation. This formalization and transparency will increase the likelihood that assumptions are properly identified and noted in the model's performance and that the correct model will be chosen and used for the specific application.

ACKNOWLEDGEMENTS

The authors would like to thank Alion Science and Technology who completed the MIDAS 5.x augmentations under NASA's technical guidance, Sandra G. Hart who was instrumental in guiding the MIDAS transparency augmentations, and all reviewers of the current paper for their insightful comments. The MIDAS 5.x augmentations discussed in the present paper were funded by the NextGen-Airspace project, and the paper composition was funded by the NextGen-Airportal and AvSAFE IIFDT. For more information on MIDAS 5.x, the reader is asked to consult the MIDAS web site at http://hsi.arc.nasa.gov/groups/midas/.

REFERENCES

Boring, R.L., Dudenhoeffer, D.D., Hallbert, B.P., & Gore, B.F., 2006, Virtual power plant control room and crew modeling using MIDAS. Proceedings of the joint Halden reactor project and CSNI special experts' group on human and organisational factors workshop on future control station designs and human performance issues in nuclear power plants, May 8 – 10, 2006, Halden, Norway, pp. 1 – 5:5.3.

Corker, K.M. & Smith, B.R., 1993, An architecture and model for cognitive engineering simulation analysis: Application to advanced aviation automation. AIAA Computing in Aerospace 9 Conference . October, 1993: San Diego, CA.

Fitts, P. M., 1954, The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology,47, 381–391.

Gluck, K.A. and Pew, R.W. (Eds.), 2005, *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation and validation*, New Jersey: Lawrence Erlbaum & Associates.

Gore, B.F., 2002, An emergent behavior model of complex human-system performance: An aviation surface related application. VDI Berichte 1675, 313-328. Düsseldorf, Germany: VDI Verl..

Gore, B.F. 2008, Chapter 37: Human performance: Evaluating the cognitive aspects. In V. Duffy (Ed.), *Handbook of digital human modeling*, NJ: Taylor and Francis.

Gore, B.F. & Corker, K., 2000, System interaction in free flight: A modeling tool cross- comparison. SAE transactions - journal of aerospace, 108 (1), 409-424, Warrendale: SAE Inc.

Gore, B.F. & Smith, J.D., 2006, Risk assessment and human performance modeling: the need for an integrated approach. In K.A. Malek (ed.) *International Journal of Human Factors of Modeling and Simulation*, 1(1), 119-139.

Hamilton, D.B., Bierbaum, C.R. & Fullford, L.A., 1990, Task analysis/workload (TAWL) user's guide, version 4 (Research Project 91-11). Alexandria, VA: US Army Research Institute for the Behavioral and Social Sciences (AD A241 861).

Hart, S. G., & Staveland, L. E., 1988, Development of NASA-TLX (task load index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.) Human mental workload. North-Holland: Elsevier Science Publishers (pp. 139-183).

Hooey, B. L. & Foyle, D. C., 2008, Advancing the state of the art of human performance models to improve aviation safety. In D. C. Foyle, & B. L. Hooey (Eds.), Human performance modeling in aviation. Boca Raton: CRC Press/Taylor & Francis.

Leiden, K. & Best, B., 2008, A Cross-model comparison, In D. C. Foyle & B. L. Hooey (Eds.) Human Performance Modeling in Aviation. Boca Raton: CRC Press/Taylor & Francis

McCracken, J.H. & Aldrich, T.B., 1984, Analysis of selected LHX mission functions: Implications for operator workload and system automation goals. (Technical note ASI 479-024-84(b)), Fort Rucker, AL: Anacapa Sciences, Inc.

Miller, C., 2000, The human factor in complexity. In Automation, control and complexity: New developments and directions, ed. T. Samad and J. Weyrauch. New York, NY: John Wiley.

Napierski, D.P., Young, A.S. & Harper. K.A., 2004, Towards a common ontology for improving traceability of human behavior models. In the Proceedings of the 9th annual behavioral representation in modeling and simulation conference.

National Aeronautics and Space Administration (NASA), 1989, Human performance models for computer aided engineering. In Panel on pilot performance models in a computer-aided design facility, eds. J.I. Elkind, S.K. Card, J. Hochberg, & B.M. Huey. Washington D.C.: National Academy Press.

NextGen, 2007, Concept of operations for the next generation air transportation system, Version 2. Joint Planning and Development Office, Washington, DC: JPDO.

Pew, R.W., Gluck, K.A., & Deutsch, S., 2005, Accomplishments challenges, and future directions for human behavior representation. In K.A. Gluck, and R.W. Pew (eds.) (2005). *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation and validation, (pp.397-414),* New Jersey: Lawrence Erlbaum & Associates.

Seppelt, B.J. & Lee, J.D., 2007, Making adaptive cruise control (ACC) limits visible. Int. J. Human-Computer Studies, 65, 192–205

Sheridan, T.B., 1992, Telerobotics, automation, and human supervisory control. MIT Press: Cambridge.

Sheridan, T. B., & Ferrell, W.R., 1974, Man-machine systems. Cambridge, MA and London, England: The MIT Press.

Wickens, C.D., 1984, The multiple resource model of human performance: implications for display design. In: Proceedings of AGARD/NATO Report, Williamsburg, VA: AGARD/NATO.

Zhai, S, & Milgram, P., 1991, A telerobotic virtual control system. In Proceedings of SPIE Volume 1612, Cooperative intelligent robotics in space II, ed. W. Stoney. Boston: SPIE.

Zhai, S, & Milgram P., 1992, Human-robot synergism and virtual telerobotic control. In Proceedings of the annual meeting of human factors association of Canada. Canada: Human Factors and Ergonomics Society.