Abstraction of analytical models from cognitive models of human control of robotic swarms

Katia Sycara[1], Christian Lebiere[1], Yulong Pei[1], Don Morrison[1], Yuqing Tang[1], and Michael Lewis[2]
[1]Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, Pennsylvania, USA
{katia,cl,yulongp,dfm2,yuqing.tang}@cs.cmu.edu
[2]University of Pittsburgh, 4200 Fifth Avenue
Pittsburgh, Pennsylvania, USA
ml@sis.pitt.edu

Abstract

In order to formally validate cyber-physical systems, analytically tractable models of human control are desirable. While those models can be abstracted directly from human data, limitations on the amount and reliability of data can lead to overfitting and lack of generalization. We introduce a methodology for deriving formal models of human control of cyber-physical systems based on the use of cognitive models. Analytical models such as Markov models can be derived from an instance-based learning model of the task built using the ACT-R cognitive architecture. The approach is illustrated in the context of a robotic control task involving the choice of two options to control a robotic swarm. The cognitive model and various forms of the analytical model are validated against each other and against human performance data. The current limitations of the approach are discussed as well as its implications for the automated validation of cyber-physical systems.
Keywords: Cyber-physical systems; ACT-R cognitive models; Markov models; Robotic control

Introduction

As robotic platforms become more robust, teams of autonomously coordinating robots (robotic swarms) may be deployed for various tasks including environmental exploration, large-scale search and rescue, border protection, etc. One of the most important challenges in the design and deployment of such systems is making them amenable to effective human control. This requirement is complicated by the nonlinear dynamics of robotic swarm systems, the need to make realistic environmental assumptions, and the limitations and capabilities of human cognition. There has been much recent interest and research activity in control theory for formal system verification of safe operation of automation. In such work either the human has not been modeled at all, or the human has been modeled as a system disturbance. Modeling mixed human-autonomous systems where human cognition is taken into account is in its infancy. Formal and validated models of human-autonomous systems' safe operation, where the human element is modeled realistically, would be beneficial not only because these models would provide guarantees of performance, but also because they may uncover parts of the control space where human performance can deteriorate to unacceptable levels. Human cognitive limitations, the nonlinearity of the state-evolution dynamics of autonomously coordinating robots, and the high dimensionality of the joint state space of such systems preclude the possibility of a human maintaining or predicting the joint state of the whole system. Furthermore, the human may perform a broad spectrum of tasks ranging from reactive tasks, like manual control, to high-level deliberative tasks, like taking go/no-go decisions for a particular sub-mission. Cognitive modeling based on cognitive architectures such as ACT-R (Anderson & Lebiere, 1998; Anderson, 2007) has existed for many years. However, the resulting models are not in a mathematical form that is amenable to the techniques of formal verification. One way of meeting this challenge is creating an analytic model of human performance based on a cognitive model. Such an analytic model is cognitively compatible by construction, and because of its mathematical nature, is in the appropriate form for formal verification. In the case of a human operator controlling a robotic swarm, the analytic model can be integrated with a formal model that describes the swarm dynamics so that the overall mixed human-swarm system can be formally verified.

This paper presents the methodology of development of such an analytic model based on an ACT-R cognitive model. The task for which the cognitive model and the analytic model were constructed was the control of a robotic swarm simulation. The analytic model development process starts with data from human experiments. Human-in-the-loop experimentation supports the development and validation of descriptive cognitive models in two stages. Initial development and data collection from the simulation are used to bound expected performance and familiarize experimenters with the domain and its issues, as well as to constrain the task-independent control model to reflect general procedures. The data provides the experience needed to train the model using the Instance Based Learning (IBL) methodology (Gonzalez, Lerch, & Lebiere, 2003) in order to generate appropriate knowledge representations in memory in the form of control instances that guide decisions, as well as to tune general architectural parameters that modulate performance. Attentional routines can also be integrated to represent limitations in the speed and capacity of processing information in complex situations. Instances are grounded in specific situations, making them easy to learn through direct experience with the system in an automated process sometimes called chunking. Instances generalize dynamically to similar situations, providing predictions of performance in not previously experienced or partially experienced situations and resulting in situation-specific representations in short-term memory.

In building a (stochastic) state space model of a human the primary challenges are defining the relevant states and transforming the human constraints in the neuroscience and psychology literatures into state space constraints. We use the cognitive model as a proxy for the human operator and run simulations to produce the decisions made by the model as a function of operator cognitive state and cognitive limitations. The methodology and resulting model will be described in detail in the rest of this paper.

Experiment Task

The human-swarm system studied followed that described in (Bullo, Corts, & Martinez, 2009). Participants control a swarm of twenty simulated robots in a web interface. No control can be exerted over individual robots, only over the swarm as a whole, and only by the choice of one of two strategies controlling how the robots collectively move: Rendezvous or Deploy. The two strategies correspond to two different algorithms for the evolution of the robots' motions, Rendezvous causing the positions to largely converge, and Deploy to largely diverge. In addition to the robots themselves, the simulated environment also contains a set of fixed obstacles. Each of the sixty trials begins with a set of initial positions of the robots, and of the obstacles. These positions were sampled from bivariate Gaussian distributions, a different pair of distributions used at each trial. The means and variances of these distributions were themselves samples from a uniform distribution. While each participant saw roughly the same sets of positions, in the same order, a small amount of noise was introduced into each.

The interface presents the initial positions of the robots and the positions of the obstacles, and solicits a choice of Rendezvous or Deploy from the human. The robots then move according to that strategy, and leave a visual trail of where they have been (Figure 1). The interface also displays direct feedback in the form of a number representing the percentage of the environment's area that the ensemble of robots has covered. The human's goal in each trial is to select the strategy that can be expected to result in the larger coverage, for that set of initial robot positions and obstacles.

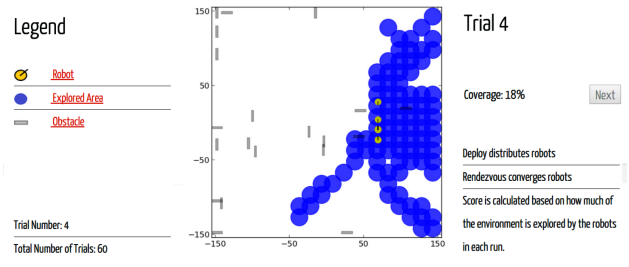

Figure 1: The interface to the simulated swarm experiment.

Fifty participants were recruited within Amazon Mechanical Turk[1], of whom forty-eight completed the experiment. Each of the forty-eight participants was presented with sixty trials. The first ten were training trials. In these training trials participants were asked to choose Rendezvous or Deploy and observe the resulting coverage. They were then asked to choose the forgone strategy and see its resulting coverage. After the training trials they were only able to select one of the strategies, and saw only the coverage result for the chosen strategy, with no feedback on the forgone strategy. Each of the first thirty post-training trials were unique, but the last twenty were alternately unique, and recapitulations of the training trials, modified slightly by the addition of noise. The participants were not told that there would be such recapitulated trials.

[1] https://www.mturk.com/mturk/welcome

Cognitive Model

The cognitive model is implemented in the neurally-inspired cognitive architecture ACT-R and follows the instance-based learning (IBL) methodology. In IBL decisions are made primarily based on experiences of a task. In our model these experiences are stored in the chunks of ACT-R's declarative memory, each such chunk corresponding to a relevant experience. Instance chunks typically contain a description of the context in which each decision is made, the decision itself, and the outcome of that decision. The mean initial position of the robots (eccentricity) and its variance (dispersion) were used to characterize the context of each trial. Other dimensions were considered, especially characterizing the distribution of obstacles, but were found upon further analysis both relatively inconsequential to the outcome and generally ignored by human participants. Both possible decisions were represented in each instance chunk, with the outcome in terms of coverage for each action stored in dedicated slots. Each instance chunk therefore contains four slots mapping the eccentricity and dispersion of the robot swarm to the coverage percentage for the Rendezvous and Deploy actions.

Before the model proper begins executing, chunks containing the ground truth values of both the Rendezvous and Deploy values for each of the ten training trials are added directly to declarative memory. For each of the fifty non-training trials of the experiment the model is presented with the eccentricity and dispersion values, and estimates, from the chunks stored in declarative memory, expected coverage fractions for Rendezvous and Deploy. These estimates are generated using ACT-R's blending mechanism (Lebiere, 1999), using partial matching of the chunks representing instances that are already in memory. This partial matching is done using a linear similarity function between the eccentricity and dispersion values, as well as the usual ACT-R declarative memory retrieval's activation computation, including recency and noise. The model selects as its decision whichever of the two actions produces the larger expected coverage percentage. The model then receives as feedback the ground truth coverage for the chosen Rendezvous or Deploy action, and registers it instead of its estimate in its representation of the current trial. The coverage value in this chunk for the forgone

option is, instead of the ground truth, left as the model's estimate. Upon completion of the trial the representation of the problem is added as a new chunk in declarative memory. The model thus starts out with ten instances, those from training, and builds up to sixty by the conclusion of the experiment as its experiences accumulate.

The ACT-R model is stochastic, and was run 1,000 times to generate stable estimates, each with a distinct random number seed. Most ACT-R parameters were left at their default values. The main deviation from standard values was to set the activation noise parameter to a relatively high value of 0.75 to reflect the high stochasticity of decisions made by the Mechanical Turk subjects. For each run declarative memory is reinitialized with just the ten training trials, and the full set of sixty instances is built up afresh, with potentially different values in them reflecting the stochasticity of the model's judgment at each step, and most specifically the fact that it receives feedback on only its chosen option and its potential implications for the dynamics of its behavior ((Lebiere, Gonzalez, & Martin, 2007)). The results are aggregated both for comparison to the human results and for constructing the Markov model.

Abstraction Procedure

The knowledge state in IBL models is characterized by the set of instance chunks and their activation. The evolution of the cognitive state as the model accumulates experiences can be thought of as a k-dimensional discrete-time signal, which is the time-trajectory of activation levels of the different memory chunks through various decision cycles, in response to particular inputs.

As stated before, an IBL memory chunk in ACT-R consists of a representation of the context and outcome of the control actions. In this setting, context involves the centrality and dispersion of the robots, while the outcome involves a representation of the percentage coverage achieved by the available decisions. Environment and system observations change the activation levels of the existing memory chunks as well as add new chunks to the model, thus reflecting the system state as observed by the operator. Abstracting that distributed state of knowledge contained in the cognitive model's declarative memory in an analytical model requires coarsening it into discrete states, such as the degree of preference toward one strategy or the other. To reflect the context-sensitive nature of the IBL decision process, distinct sets of states are created for each context neighborhood. The number and nature of the states is left to the modeler. The changes that each experience causes to the activations (and number) of instance chunks in memory are reflected in a probabilistic transition in the analytical model. The transitions in the analytical model are trained from Monte Carlo runs of the cognitive model.

For this specific model, our approach starts with an interface to the multi-robot system that explicitly represents two actions, "Deploy" and "Rendezvous", and the percentage coverage obtained by the action. In this setting, the decision context involves the centrality and dispersion of the robots. The memory chunk also contains a representation of the action taken as well as the action not taken, and the resulting and expected outcome, respectively. Environment and system observations change the activation levels of the memory chunks in the cognitive model, thus reflecting the system state as observed by the operator. Previously created chunks decay with time and are reinforced with retrieval, while new chunks are added to reflect recent observations. The time-trajectory of the activation levels of the memory chunks can be clustered to produce the Markov model. States of the Markov model are defined to correspond to a pattern of memory chunk activation levels. In this domain, they would correspond to a temporary preference for one action over the other. In general there can be k events that correspond to the consistent states of the system as observed by the operator.

Markov Model

Following the approach described in (Gray, 2002), we employ a Markov model as the analytic model for ACT-R cognitive processes of human control. Let D denote the selection of Deploy and R denote the selection of Rendezvous. Specialized for the human-swarm task, the overall Markov model of the cognitive processes is decomposed into two sub-Markov Models indicated by superscripts in the edges of the graphs that correspond to two basic outcomes of the action chosen (see right-hand side of Figure 2): a) Model U: the ground truth coverage is larger than the estimation of the ACT-R model; b) Model L: the ground truth coverage is less than or equal to the estimation of the ACT-R model. Model U is parameterized by four probabilistic action selection transitions: 1) $p^U_{D\to R}$, 2) $p^U_{D\to D}$ (where $p^U_{D\to R}+p^U_{D\to D}=1$); 3) $p^U_{R\to D}$, and 4) $p^U_{R\to R}$ (where $p^U_{R\to D}+p^U_{R\to R}=1$). Symmetrically, Model L is also parameterized by four probabilistic state transitions: 1) $p^L_{D\to R}$, 2) $p^L_{D\to D}$(where $p^L_{D\to R}+p^L_{D\to D}=1$); 3) $p^L_{R\to D}$, and 4) $p^L_{R\to R}$ (where $p^L_{R\to D}+p^L_{R\to R}=1$). The switching between these two sub-models is parameterized by the probabilities $p_{Grd>Est}$ and $p_{Grd\leq Est}$ where $p_{Grd>Est}+p_{Grd\leq Est}=1$ and where Grd is the ground truth and Est is the ACT-R estimation. As a result, we establish a Markov model of action selection assuming that any action selection is independent of the history given the previous action and the chosen sub-model (either U or L).

To situate the Markov model in the human-swarm environment, we discretize the observation space (the dispersion and eccentricity) as a grid of cells (see the left-hand side of Figure 2). Each cell in the grid is associated with an overall Markov model as described above.

After the ACT-R model generates the data, the training and prediction test procedure is as follows: (1) locate the cells to which the data instances belong; (2) train a Markov model for each cell; (3) make predictions for each instance in test data based on the Markov transition probabilities.

Training Procedure

Situated in the grid of the environment, a Markov model (see the right-hand side of Figure 2) is trained for each cell.

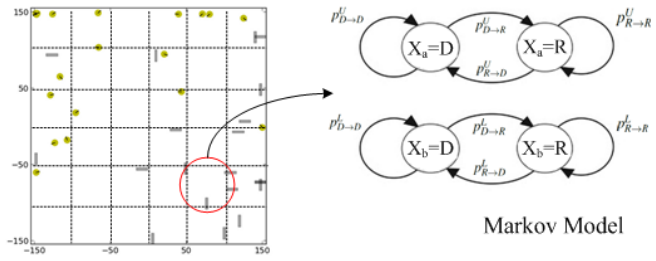

Figure 2: The framework for the training procedure in the Markov model.

The nodes $X_{a/b} = D$ indicate the selection is Deploy and the nodes $X_{a/b} = R$ indicate the selection is Rendezvous. In each cell, the data is represented as a sequence of selections $X = \{x_1, ..., x_t\}$. During the training procedure, scanning through the sequences of action selections X, we record the counts of action selection transitions (x_i, x_{i+1}). Formally, the count $c^k_{s_i \to s_j}$ is the number of transitions from selection s_i to s_j ($s_i, s_j \in \{D, R\}$) of the sub-model k (where $k \in \{U, L\}$). For example, when an action selection transition $(D, R) \in X$ is encountered — the current selection is Deploy and the next selection is Rendezvous — and the feedback is that the ground truth is less than the estimation (in sub-model L), then the count is updated as: $c^L_{D \to R} \leftarrow c^L_{D \to R} + 1$. After counting all the instances in the data set, these counts are normalized into the transition probabilities following:

$$p^k_{s_i \to s_j} = \frac{c^k_{s_i \to s_j}}{\sum_{s_t \in \{D,R\}} c^k_{s_i \to s_t}}. \tag{1}$$

In addition, the switching probability $p_{Grd>Est}$ ($p_{Grd \leq Est} = 1 - p_{Grd>Est}$) between the sub-models is estimated by simply computing the ratio of the times that the ground truth coverage is larger than the estimation of ACT-R over the times that the ground truth coverage is less than or equal to the estimation of ACT-R. It can be proved that the above parameter estimation process computes the model parameters that maximize the posterior probability of generating the data conditional on the parameters.

Prediction Procedure

After the training procedure, we obtain the overall Markov model, which is characterized by the probabilities from Deploy to Deploy $p_{D \to D}$, from Deploy to Rendezvous $p_{D \to R}$, from Rendezvous to Deploy $p_{R \to D}$ and from Rendezvous to Rendezvous $p_{R \to R}$, as a switching mixture of the two sub-models (Model U and Model L):

$$p_{D \to D} = p_{Grd>Est} * p^U_{D \to D} + p_{Grd \leq Est} * p^L_{D \to D} \tag{2}$$

$$p_{D \to R} = p_{Grd>Est} * p^U_{D \to R} + p_{Grd \leq Est} * p^L_{D \to R} \tag{3}$$

$$p_{R \to D} = p_{Grd>Est} * p^U_{R \to D} + p_{Grd \leq Est} * p^L_{R \to D} \tag{4}$$

$$p_{R \to R} = p_{Grd>Est} * p^U_{R \to R} + p_{Grd \leq Est} * p^L_{R \to R}. \tag{5}$$

This overall model can be exploited to predict the next action selection s_{i+1} of the human players given the current action selection s_i following the decision rule:

$$s_{i+1} = \arg\max_{x \in \{D,R\}} p_{s_i \to x}.$$

For example, if the overall model states that $p_{D \to D} > p_{D \to R}$, and the current selection is Deploy, the predicted next action will be Deploy; otherwise, the next prediction is Rendezvous.

Results

The resulting Markov model is evaluated by two measures: **Accuracy** and **MSE** (Mean Square Error). The **Accuracy** is defined as:

$$Accuracy = \frac{|\mathbb{I}(SEL_{\text{pred}}, SEL_{\text{ACT-R}})|}{|\text{trials}|} \tag{6}$$

where $\mathbb{I}(SEL_{\text{pred}}, SEL_{\text{ACT-R}}) = 1$ if the prediction selection is the same as the ACT-R selection; otherwise $\mathbb{I}(SEL_{\text{pred}}, SEL_{\text{ACT-R}}) = 0$. And **MSE** is defined as:

$$MSE = \frac{1}{|\text{trials}|}(P_{\text{Markov-Grd}} - P_{\text{ACT-R-Grd}})^2 \tag{7}$$

where $P_{\text{Markov-Grd}}$ is the precision of the Markov model (i.e. the Markov model conforms with the ground truth) and $P_{\text{ACT-R-Grd}}$ is the precision of the ACT-R model (i.e. the ACT-R model conforms with the ground truth).

We evaluate our Markov model given the following discretization of the observation space: 17×17, 10×10, 5×5, 3×3 and 1×1. The model prediction performance is shown in Table 1. From Table 1, we can see that the performance improves as the granularity of the grid is increased but reaches a plateau around a 5×5 grid, which is a plausible discretization level. The limit on accuracy of about 75 percent fundamentally reflects the variability of human decisions. The limit on mean square error fundamentally reflects the discretization of the problem space and other factors averaged over by the Markov model training procedure.

Number of cells	Accuracy	Mean Square Error
17×17	75.07%	0.06718
10×10	74.90%	0.07671
5×5	74.48%	0.08449
3×3	69.61%	0.11254
1×1	52.99%	0.29963

Table 1: The prediction results of different numbers of cells.

Figure 3 presents the trial-by-trial performance of the human participants, the cognitive model, and three versions of the analytical model using various degrees of state coarseness. The cognitive model generally captures quite well the pattern of fluctuations of human performance across trials. The fluctuations reflect both the impact of previous outcomes,

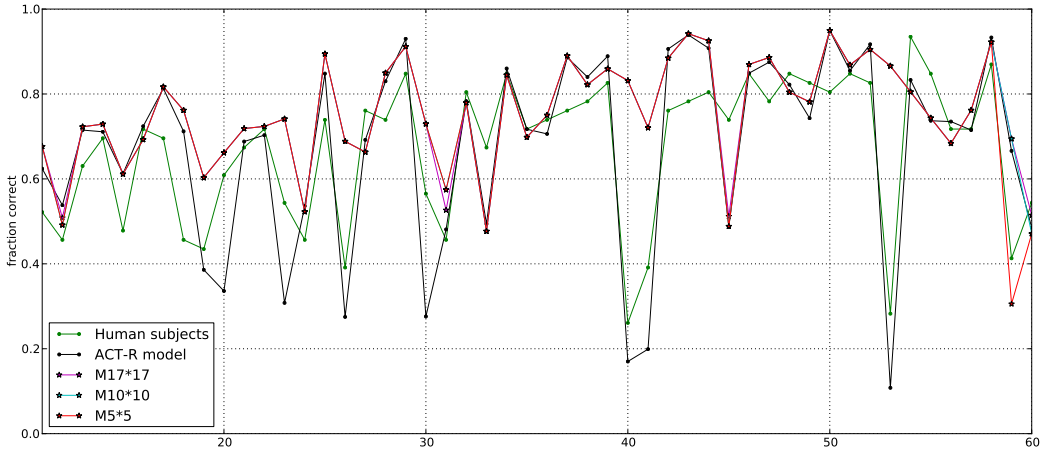

Figure 3: Fraction of decisions correct by trial number

which are captured in the cognitive model by the addition of a new chunk for each experience as well as the recency effect from activation decay, and the effect of each new trial context on the decision. The analytical models show occasional deviations from that pattern, which reflects the coarse nature of their state space and other simplifying factors. A learning effect can be observed in the increasing consensus across runs for each action choice, whether correct or incorrect.

Figure 4 graphs the fraction of choice of a specific option (Rendezvous) as a function of the difference in coverage between that option and the alternative (Deploy) in the ground truth data. The sharp sigmoid curve centered around the origin fit to the data indicates that both human participants and cognitive model learn to perform the task quite well, and nearly identically. Their errors primarily reflect contexts in which the two actions provide very similar performance. The analytical models are also sensitive to differences in coverage, but not nearly as sharply as their sigmoid fits are much flattened. This presumably reflects the coarse state representation that aggregates nearby contexts in identical bins as opposed to the more graded similarity-based partial matching of the cognitive model. In addition, when limited to 3x3 cells, the analytical model shows an inability to converge to the same certainty as the cognitive model for large differences in coverage.

Figure 5 graphs the pattern of choices in the two-dimensional context space of robot dispersion (x-axis) and eccentricity (y-axis). Green circles are associated with a correct choice of Rendezvous, and are typically associated with large dispersion values, while yellow circles are associated with a correct choice of Deploy, and are typically associated with small dispersion values. The size of the circle represents the probability of choosing the correct action. Larger choice probabilities are typically seen for extreme dispersion values, while smaller probabilities are seen for mid-range dispersion

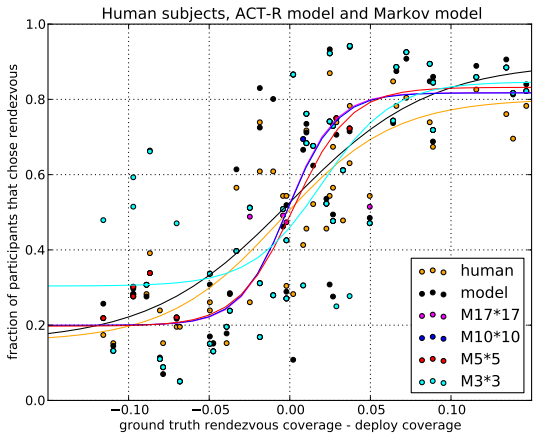

Figure 4: Fraction choice of the Rendezvous action as a function of the difference in percent coverage between Rendezvous and Deploy

values that correspond to the boundary between the two domains where the difference between the two actions is small. For each trial, circles centered on the same point are plotted for both human and cognitive model choices. Most pairs of circles overlap perfectly but specific discrepancies between human and model choice are visible, corresponding to trials 18, 20, 23, 30, 41, 45 and 59 (see corresponding data on Figure 3). All those trials are located in the boundary region where small differences in perception or experience might easily make the difference between choosing one action over the other.

Conclusion and Future Work

This approach can be understood as one of incremental abstraction in model development. We start with the full detail

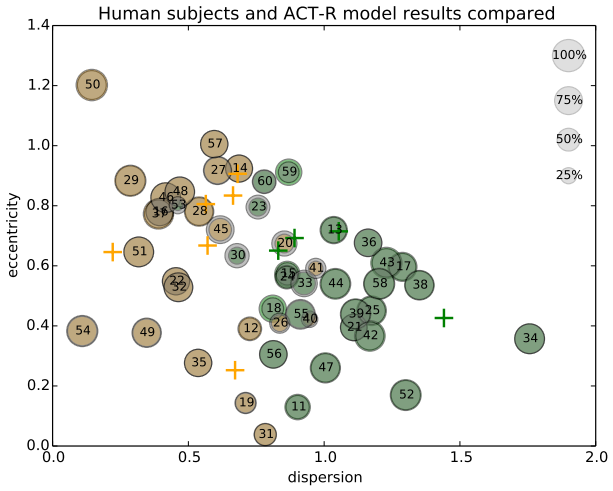

Figure 5: Performance of human participants and cognitive model (circles) graphed in the two-dimensional context space for Rendezvous (Green) and Deploy (yellow) trials. Crosses indicate training trials.

of the human data. In this case, it included only the choice between two competing actions. However, it could also include latency to make the decisions, individual variations, or any other observable relevant aspect of human performance. A process model of human control of the cyber-physical system is then developed using a cognitive architecture. The benefit of using a cognitive architecture is that it already includes many constraints on performance that do not need to be rederived from data. A further advantage of using the IBL modeling methodology is to further limit modeler choices and improve the automated nature of this approach by limiting modeler decisions to the representation of the context. Further abstraction can then be achieved by specifying the structure of a formal analytical model, such as the cells and states of the Markov model used here. Unlike limited and noisy human data, the cognitive model can then be run as many times as needed and the resulting data used to train the formal model to the needed accuracy. The models can be validated against each other and against the human performance data at each level of development: (a) the initial cognitive model can be compared to the human data that it is meant to capture, (b) the formal analytical model can be compared to the cognitive model from which it is abstracted, and finally (c) the formal analytical model can be validated against the human performance data.

One important question is which aspects of the cognitive model performance can be readily captured by this approach? We saw that in this domain the model's Markovian assumption was quite accurate at capturing the impact of experience on decisions. Coarsening the high-dimensional nature of declarative memory representation into a limited number of states can lead to some distortions but seems fairly accurate if the state space is above some minimum threshold. Another limitation is the need to restrict contextual generalization to an all-or-none division into independent cells. Another important question is how to generalize the Markov model for more realistic applications which have 1) larger action space (more than 2 actions), 2) higher dimension of the observation space (more than 2 observed parameters), and 3) more sophisticated performance dependency over action selections and environment observations.

Finally, the analytical model is trained on the entire data set generated by the cognitive model, including the model's initial learning curve. As it is, the analytical model is akin to a representation of average or asymptotic performance. More contextual elements would have to be added to enable a representation of cognitive learning processes in the analytical model.

Our future work involves two parallel thrusts. We want to generalize our approach to modeling human control of other cyber-physical processes to test its breadth of applicability. Also, we need to incorporate the resulting analytical models into formal verification frameworks, e.g. (Oishi, Mitchell, Bayen, & Tomlin, 2008), that can be used to derive formal guarantees on the human control of cyber-physical systems.

Acknowledgments

This work is funded by NSF awards CNS1329986, CNS1329762 and CNS1329878 to Katia Sycara, Michael Lewis and Christian Lebiere.

References

Anderson, J. (2007). *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press.

Anderson, J., & Lebiere, C. J. (1998). *The atomic components of thought*. Mahwah, N.J.: Erlbaum.

Bullo, F., Corts, J., & Martinez, S. (2009). *Distributed control of robotic networks: A mathematical approach to motion coordination algorithms*. Princeton University Press.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*, 591–635.

Gray, R. (2002). markov at the bat: A model of cognitive processing in baseball batters. *Psychological Science*, *13*(6), 542–547.

Lebiere, C. (1999). The dynamics of cognitive arithmetic. *Kognitionswissenschaft*, *8*(1), 5–19.

Lebiere, C., Gonzalez, C., & Martin, M. (2007). Instance-based decision-making model of repeated binary choice. In *Proceedings of the 8th international conference on cognitive modeling*. Ann Arbor, MI.

Oishi, M., Mitchell, I. M., Bayen, A. M., & Tomlin, C. J. (2008). Invariance-preserving abstractions of hybrid systems: Application to user interface design. *Control Systems Technology, IEEE Transactions on*, *16*(2), 229–244.