Influencing versus Informing Design, Part 2: Macrocognitive Modeling

Robert R. Hoffman

Vol. 23, No. 6
November/December 2008

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.
The human mind is like a black box—no one can directly observe or experience others’ mental events. So, in any sort of flow diagram linking stimulus to response, at least one box remains unfillable. It has gone by many names—among them ego, apprehension, knowing, will, consciousness, the “executive,” and reasoning. Psychologists and philosophers have labored for centuries to build a reliable foundation for a good (or “objective”) view into the black box. Modern psychologists go to extreme lengths to design and execute supremely clever controlled laboratory experiments and procedures that provide us with glimpses into the box. What’s needed in many venues, including human-centered computing for intelligent systems, is a fast track into that black box.

Software engineering is perhaps such a venue. A software engineer might start with the outputs of a documentation analysis, correctly apply user interface standards and conventions, and then produce an artifact that’s exciting to look at. However, the artifact (and the work method it instills) might not be informed by the actual work context. The previous essay in this department discussed this gap in the systems-engineering process. That essay illustrated how the gap has been successfully crossed in one direction, in a project in which cognitive systems engineers expressed the requirements in a way that captured key functionalities and the rationale for those functionalities, thereby “speaking to” the software engineer.

This essay works in the other direction: providing systems engineers with an easy-to-use method—the Macro-cognitive Modeling Procedure (MMP)—that might enable them to ramp up their understanding of the cognitive work.

Macrocognition

Previous essays in this department have discussed the distinction between microcognition and macrocognition and its implications for intelligent systems. Microcognitive and macrocognitive models take different forms and have different purposes. Microcognitive models present causal-chain understandings of mental events, built from mental operations such as short-term memory access and attentional shifts. Such laboratory-based models help us describe performance for known, fixed tasks. On the other hand, macrocognitive models describe the major goal-directed functions of cognitive work (deciding, re-planning, sensemaking, problem detection, and so on) and the cognitive processes that support those functions (for example, developing mental models and maintaining common ground). Macrocognitive modeling is likely to be based on studies of the actual work and expert reasoning in complex and dynamic situations.

The creation of macrocognitive models of aspects of sociotechnical work systems could help systems and software engineers as well as cognitive systems engineers develop high-level understandings of the nature of the cognitive work. Through a deep, rich understanding of the actual cognitive work, we can increase the likelihood of creating human-centered technologies and work methods.

We can achieve an understanding of cognitive work through several broad cognitive task analysis methodologies:
Using largely ethnometric methods, we can study the workplace and work patterns and conduct documentation analysis. The general approach is called activity analysis or work analysis.

Using largely psychometric methods, we can measure human performance and conduct cognitive task analysis. This is the general approach of human factors engineering and cognitive systems engineering.

Using largely sociometric methods, we can interview domain practitioners, study communication patterns, and reveal social networks within knowledge-based organizations. This is the general approach of ethnomethodology, although it overlaps significantly with activity analysis and work analysis.

Applying these methods results in macrocognitive descriptions of practitioner knowledge and practitioner reasoning.

**The Macrocognitive Modeling Procedure**

Software engineers don’t need to carve time away from system development (whether at the high level of general architectures or the detailed levels of coding) to conduct all the research that cognitive systems engineers would like to conduct in order to come to a detailed and rich understanding of the work. What software engineers need is that fast track into the black box. Just as the creation of an Abstraction-Decomposition Matrix can bridge the gap from the language of the cognitive systems engineer to that of the systems engineer, the MMP might efficiently help the systems engineer bridge the gap between his or her needs and those of the cognitive systems engineer.

My colleagues and I first conceived the MMP to support the efficient creation and subsequent validation of macrocognitive models of practitioner reasoning, thus avoiding labor-intensive protocol analysis. The MMP evolved after we had developed a general model of expert reasoning, which incorporated such processes as recognition-primed decision making, situational awareness, and mental model formation.

This general model of expert reasoning included Karl Duncker’s notion of the hypothesis-testing refinement cycle. The “base model,” shown in Figure 1, captures (as variations on a theme) a considerable number of proposed hypothetical reasoning sequences taken from studies of diverse domains of expertise. We also created a variation of this model to capture results from studies of expert weather forecasters’ reasoning (see Figure 2).

My colleagues and I first conceived the MMP when we showed a weather forecaster the Figure 2 model and asked him whether it seemed appropriate to the domain. Ordinarily, an experimental psychologist’s theoretical concoction would be foreign language to a domain practitioner. In this case we felt that discussing the model would be sensible because the weather forecasting community had for years relied on a distinction between conceptual (mental) models and computational models of the weather. The forecaster we were working with spontaneously took the diagram as an opportunity to add domain-specific details to the process description and modify some of the relationships among the diagram elements. With this experience as a flash point, we created a more formal three-step procedure for the research method.

**Step 1: Preparation**

Adapt the base model (see Figure 1) to make it directly pertinent to the domain.
Using weather forecasting—for example, comparing Figures 1 and 2—you’d specify the “problem of the day” as “the forecasting problem of the day,” and “data examination” as “examination of images, data, radar.” Next, create two alternative bogus models. At least one should include some sort of loop, and both should include some of the base model’s elements. Taken together, the bogus models include core macrocognitive functions (for example, recognition priming, hypothesis testing, and so on). Ideally, these models aren’t too unrealistic; nevertheless, expect the practitioner not to be entirely satisfied with either of them. Examples appear in Figure 3.

**Step 2: Model Making**

Show the domain practitioners (who span a range of proficiency) the bogus models, and invite them to pick the one best representing their strategy. Then, using the bogus models and their elements as a scaffold, invite each practitioner to concoct his or her own reasoning diagram.

**Step 3: Verification**

There is some likelihood that the model created in Step 2 might be a “just-so story,” that is, a description of what the practitioner says their reasoning is. After a few days (or more), place yourself in the workplace and observe each practitioner as he or she arrives and begin their day’s work. You can validate some elements of the model by observation (for example, “examine satellite images”). Other elements aren’t so readily validated, but they might be subject to probing questions (such as “What are you thinking now?” and “What are you doing now?”). Figure 4 shows example results for one practitioner model, indicating the Step 3 results. The call-out balloons indicate results of the observation and probe procedure. Our first use of this procedure, in the weather forecasting domain, resulted in models of the reasoning of seven proficient forecasters (experts and journeymen), validated by observations of actual forecasting activities. Developing and validating the models took 52 minutes on average. This is without doubt less than the time that other (widely used) methods need to reveal and verify reasoning models (for example, preparing and functionally coding a transcript of a think-aloud problem-solving protocol can take many hours). We were also able to validate the general model of forecaster reasoning that we developed in our initial documentation analysis (see Figure 2). The results also clearly showed differences in proficiency, with less-experienced forecasters relying uncritically on computer forecasts and less likely to think hypothetically and counterfactually. Also, the less-experienced forecasters were more likely to rely on a fixed sequence for inspecting the outputs of various computer models of the weather.

**Variations on the Theme**

The three steps constitute a procedure that you can adapt for individual projects. For
instance, you can insert a step between Steps 2 and 3 to create a form of socio-
gram, enabling you to discern patterns of knowledge sharing in the organization.
After the individual practitioners have cre-
ated their diagrams (Step 2), wait a couple
of weeks or even longer, and then show
each practitioner in the organization all
the models that were devised and ask them to
play a “guess who created this one” game.
This step reveals the extent to which the
practitioners have shared their knowledge
and discussed their reasoning. This step
also helps to identify individuals who pos-
sess special subdomain expertise or skills.
This additional step is perhaps more useful
to cognitive systems engineers than to sys-
tems engineers (who can skip it).

This forecasting study was a first at-
tempt, and limited in some ways, but prom-
ising enough for further investigation and
application. (You can download details on
the MMP procedure at http://ihmc.us:
16080/research/projects/CTAProtocols.)
The MMP holds promise for developing
reasoning models and for testing hypotheses
concerning reasoning models, in less time
than lab-based research methods take. But
in addition to the benefits and applications
for cognitive systems engineers, this method
is easy and efficient enough to be useful in
bridging the gap between systems engineer-
ing and cognitive systems engineering. In
just a couple of days, systems engineers
could come to a useful understanding of
the cognitive work that would benefit from
the new technologies (and work methods).
Using the MMP, systems engineers could
create macrocognitive models of the key
cognitive activities involved in the work
and, as an added benefit, gain rapport with
the domain practitioners.

The MMP is likely inappropriate for
studying some domains. But when it is ap-
propriate, systems engineers and cogni-
tive systems engineers could (and perhaps
should) conduct MMPs collaboratively.
There might be special value in such col-
laboration very early in a given system
development project. This would serve to
inform the systems engineers of the key
or high-level aspects of the cognitive work
while beginning to bootstrap the cogni-
tive systems engineers into their further
in-depth investigations using methods such
as Abstraction- Decomposition Matrix
analysis.¹

A final caution. It makes little sense
to think that any single macrocognitive
model will effectively capture practitioner
reasoning. One forecaster in the weather
project looked at his own model some eight
weeks after he had created it, and rejected
it, explaining that the original model was
no longer his preferred strategy because
the weather trends had changed. The pre-
viously preferred computer models were
no longer preferred. My colleagues and I
speculated that a domain such as weather
forecasting would require many dozens
of macrocognitive models of reasoning
strategies to present a rich and fair picture
of practitioner reasoning.⁵ Understanding
this could be extremely valuable to the
systems engineer. It could provide insight
into the cognitive work’s scope, diversity,
and flexibility. It could also illustrate ways
in which the work can’t be captured just by
using notions of tasks and fixed sequences
of activities.

Acknowledgments

This work resulted from effort conducted under
a contract from the National Technology Alli-
ice and from participation in the Advanced
Decision Architectures Collaborative Technol-
egy Alliance, sponsored by the US Army Re-
search Laboratory under cooperative agreement
DAAD19-01-2-0009.

References

1. R.R. Hoffman and S.V. Deal, “Influencing
versus Informing Design, Part 1: A Gap
2. G. Klein, B. Moon, and R.R. Hoffman,
“Making Sense of Sensemaking 2: A Mac-
croognitive Model,” IEEE Intelligent Sys-
81–85.
4. B. Crandall, G. Klein, and R.R. Hoffman,
Working Minds: A Practitioner’s Guide to
5. K.J. Vicente, Cognitive Work Analysis: To-
ward Safe, Productive, and Healthy Com-
puter-Based Work, Lawrence Erlbaum,
1999.
6. R.R. Hoffman et al., “Storm-LK: A
Human-Centered Knowledge Model for
Meeting Human Factors and Ergonomics
Soc., Human Factors and Ergonomics Soc.,
2001, p. 752.
7. R.R. Hoffman et al., “A Method for Eliciting,
Preserving, and Sharing the Knowl-
edge of Forecasters,” Weather and Fore-
chological Monographs, vol. 58, no. 270,
1945, pp. 1–113.
9. A. Newell, “Duncker on Thinking: An In-
quiry into Progress on Cognition,” A Cen-
tury of Psychology as a Science, S. Koch
and D.E. Leary, eds., Oxford Univ. Press,
1985, pp. 392–419.
10. R.R. Hoffman and L. Militello, Perspec-
tives on Cognitive Task Analysis: Histori-
cal Origins and Modern Communities of
11. J.G. Trafton and R.R. Hoffman, “Com-
puter-Aided Visualization in Meteorology,”
Expertise out of Context, R.R. Hoffman,
12. R.R. Hoffman, G. Trafton, and P. Roebber,
Minding the Weather: How Expert Fore-
casters Think, MIT Press, to be published
in 2009.
from Experts: A Methodological Analy-
sis,” Organizational Behavior and Human
decision Processes, vol. 62, no. 2, 1995,
p. 129–158.

Robert R. Hoffman is a senior research
scientist at the Institute for Human and Ma-
chine Cognition. Contact him at rhoffman@ihmc.us.

Software
Next Issue...
Mining Software Archives
www.computer.org/software