Neosymbiosis, How Humans and Software Benefit from Multi-Agent Cognitive Prosthesis

Russ Vane and Doug Griffith

General Dynamics Advanced Information Systems
1400 Key Blvd., Suite 700
Arlington, VA 22209
russ.vane@gd-ais.com and doug.griffith@gd-ais.com

Abstract

This paper discusses how Licklider's [Licklider 1960] idea of symbiosis can be implemented with software agents [Vane, Griffith 2005] that sense, model, project/predict and judge; using the latest theories of decision making [Vane 2006] and cognitive psychology [Kahneman 2002]. Neosymbiosis [Griffith 2005; Griffith, Greitzer in press] addresses how humans aided by software might overcome their respective weaknesses synergistically. This paper explores how neosymbiosis delivered via a cognitive automaton (CogBot), a network of software agents, might handle surprising emergent phenomena. Particular care will be given to whether an agent (either human or software) is surprised versus whether a CogBot-enhanced observer is surprised by emergence. Finally, it considers scenario involving a community of knowledge workers intelligently augmented with CogBots while they adapt to emergent threats while attempting to achieve organizational objectives.

Introduction

This approach explores human System 1 (Intuition) and System 2 (Reasoning) cognitive processing proposed by Kahneman (2002) with multi-agent support. We like the term cognitive prosthesis [Freedy 2006]. The paper discusses the need for encoded static knowledge with an explanation facility to process situational, emergent knowledge. Indirectly this paper considers the emergence of meaning and language, since the human and the agents will probably quickly develop an idiosyncratic interlingua.

By 2010, it is easy to imagine there could be a US watch center where a multi-agent community of knowledge workers and their factota, CogBots, can be working on adversarial reasoning problems. By reminding the reader of the rigors of adversarial reasoning, some of the disadvantages of attempting to build consensus in such a community are highlighted. One agent or several, software or human, may become aware of evidence of a significant emerging threat. How this awareness progresses through the organization will be the thought experiment discussed in this paper. It should be clear that truth counts, not consensus.

In such a scenario, three situations of organizational truth-seeking occur: a piece of evidence is received by a software agent that begins to change the pattern recognized by the watch center, a human either "sees" a pattern that is unrepresented in the current posse of agents, or a human leader introduces a change in mission to the organization. Additionally, total surprise could occur, neither the human nor agents perceive the evidential pattern until some undesired effect occurs. This total surprise will not be discussed either. The first two will be treated as emergent and are the focus of this discussion.

In the background section, the reader is exposed to our definition of surprise based on expectations, a reminder of know human cognitive shortfalls related to surprise, and our fine grained agents technology – concept automata/cobots. By discussing our definition of surprise in the next section we motivate architecture with variegated types of information, discuss cobot behaviors to mitigate surprise, and provide a simple theory of most relevant information to aid explanations. Lastly, we summarize and discuss where the implementation might fail to deliver emergent understanding.

Background

For background purposes every software agent is a conceptual automaton as described in the next few paragraphs and figure 1, A concept automaton, cobot.

A cobot is instantiated by a network request for a mission that it performs. This information includes performance parameters and resource constraints. It is bound to only one supervisor, the caller. In fact several competing cobots can attempt to answer the request. Conceptually, these mission requests are like Gilb's quantified requirements [Gilb 2005], a self-contained mission package which the cobot will use to judge its strategies (or encapsulated behaviors). The cobot will try to apply its known behaviors (design ideas) to project an outcome in terms of its abstract model. Once its model is established, sensory processing commences to attempt to keep the cobot "relevant to the current situation."

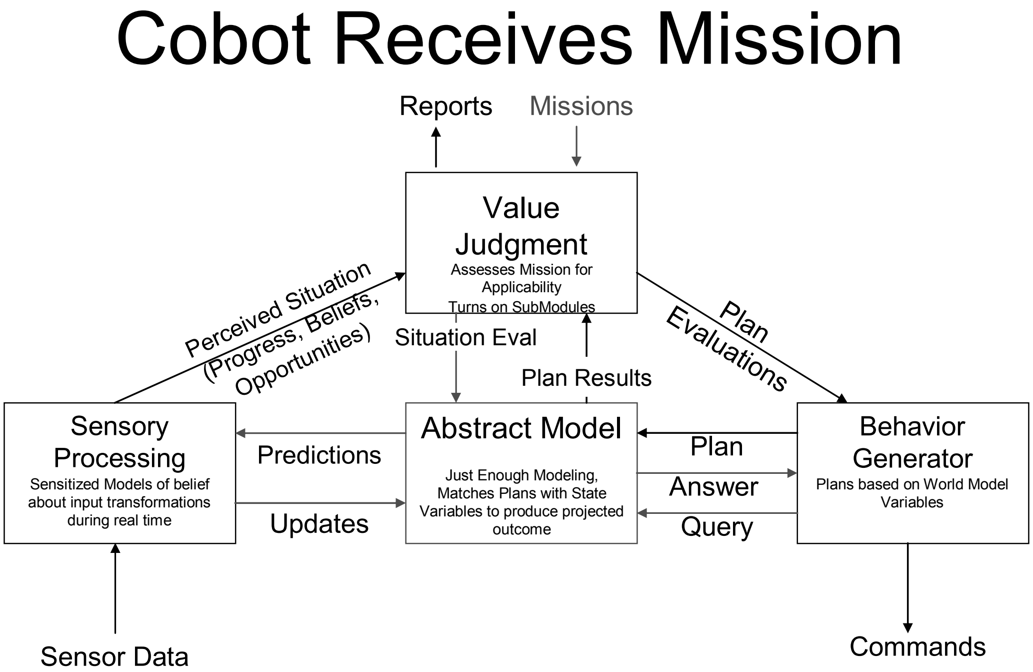

Cobot Receives Mission
Reports
Missions
Value Judgment
Assesses Mission for Applicability
Turns on SubModules
Perceived Situation (Progress, Beliefs, Opportunities)
Situation Eval
Plan Results
Plan Evaluations
Sensory Processing
Sensitized Models of belief about input transformations during real time
Predictions
Updates
Abstract Model
Just Enough Modeling, Matches Plans with State Variables to produce projected outcome
Plan
Answer
Query
Behavior Generator
Plans based on World Model Variables
Sensor Data
Commands

Figure 1, A concept automaton, cobot

Cobots can be interrupted two ways – by new information (sensor data in the bottom left) or new missions (in the top right). Sensor data is evidence about the situation. This processing is informed by "predictions" or expectations about the properties of cobot's myopic world. Uncertainty, or its opposite - confidence, can be added during this or other processing steps.

For the software agent to perceive data as evidence, three syntactic transformations need to occur:

(1) the unstructured data needs to parsed into sensed information tokens,

(2) the sensed information needs to become model relevant information, and

(3) the pedigree information must be recorded for deferred inferential tasks.

This process can fail a number of ways. For example, the input may fail to convert to usable tokens, or sufficient rules may be missing to convert it to purposeful model data, or the pedigree of the information is not available. None of these events generate surprise, they generate a paucity of usable data.

But if a cobot succeeds in capturing evidential information as updates for the model, the evidential information may be expected or not at a number of conceptual levels within a cobot. Is the data or information 'expected' or not by the agent? Is the information able to be handled by a multi-agent network with human supervisors?

A precondition for an agent to recognize a difference between what is expected and what is not, is a projection capability. In a cobot this projection comes from encoded behaviors, we prefer the term "strategies", able to be processed by the model. A model should use information about the past, present, and future to project an expectation. It should also send a prediction to the sensory processing module to condition it to handle changing/new information. Unfortunately these words are very rich and nuanced, so we would prefer to use them in their general sense. The cobot's model is its process for using evidence that it "understands" with strategies for which it has projection functions, to make plan results and predictions.

Plan results are in model terms, but not yet interpreted by the assigned mission. The judgment module converts plan results into plan evaluations (how good they are with respect to the assigned mission). GDAIS uses hypergame theory to judge the sensitivity of the plan evaluations over a number of contexts.

The network of supervisory agents assigning missions usually receives a report of the plan result, also. The plan evaluations are not sent for a number of reasons. First is that the "goodness" functions would have to be understood by all receiving cobots. These can be based on lots of local information only needed by the current cobot. For instance, if the supervisor asks for food, and the cobot responds "four portions of spaghetti primavera," not 1.78 utiles, in twenty four minutes for the assigned mission. This allows the other cobots' separate contexts to "inform" the focus cobot of constraints that it does not consider by other mission messages.

Furthermore, data must move into the structure of evidence, beyond information. Assumptions about the evidence's relationship with truth may be hidden by an immediate recording of the informational parts in a data store. As Schum writes so clearly [Schum 2005]; accuracy, credibility of sources, inference mechanisms, ALL impinge whether any agent in the organization believes the information. The cobots implement these inference mechanisms in this system.

Known Human Cognitive Shortfalls

Humans are able to be surprised also. The following three paragraphs give anecdotal evidence of how unaided (or poorly augmented) humans will make mistakes that lead to cognitive errors and poor performance. While "extreme luck" is a possible context which might mitigate some effects of these errors, it is ignored in this discussion.

First and foremost is confirmation bias. At some point while thinking, we humans classify the situation as a certain kind or type with specific properties. This allows us to focus, but also encourages us to discount almost all evidence that we receive about what is actually happening that does not conform. Lost elections, lost campaigns, and financial ruin are significant results of this kind of error. Classically, temporary losses that are managed as sub-elements of a larger context are considered not surprising; such as having some bad poll numbers during the election campaign, or losing a battle or fort during a military campaign, or having a few losers in a well-balanced portfolio. This kind of wisdom and understanding can make surprise very hard to judge and can prolong the confirmation bias.

Humans can suffer from sensory overload, and badly designed computer systems have exacerbated this problem. Human coping strategies vary enormously; and can even lead to catatonia or playing "free cell." We want our system of cobots to summarize the key features of any scenario so that humans can set the right policy or discover the key (highly-leveraged) information in the ocean of data. This filtering, requires trust between agents and a quick, cogent explanation facility to lessen confusion (human or cobot).

Finally, for this section, humans can be innumerate. Humans often do not do arithmetic well and introduce errors into calculations with fractions and other decimal representations. The software agents are ideal at these calculations.

A Theory of Explanation Priority

Vane and Griffith have presented a theory of explanation [Vane, Griffith 2006] which presupposes a human-like priority of response. By answering "why" questions given to the system, he proposed to prioritize answers based on the maximum change in computed value based on plausible contexts. Plausibility is judged by a distance measure from the current bounding contexts, C_k. That is by reviewing the value of what is expected by associated "Near contexts", the first "why" answer is the biggest change in value (positive or negative. The answer to "Why do you have an umbrella is based much less on a change in the day of the week than on "possible rain" or "blazing sun" contexts.

Explanation = $\max(\frac{\Delta V_i}{\Delta C_k})$, where V_i is the value of option i and C_k is the k^{th} context.

Discussion

This section introduces *hypergame theory* as a surprise control mechanism and uses cobots to clarify kinds of surprise, and their consequences, to CogBot networks. A table of possible surprises is carefully explained.

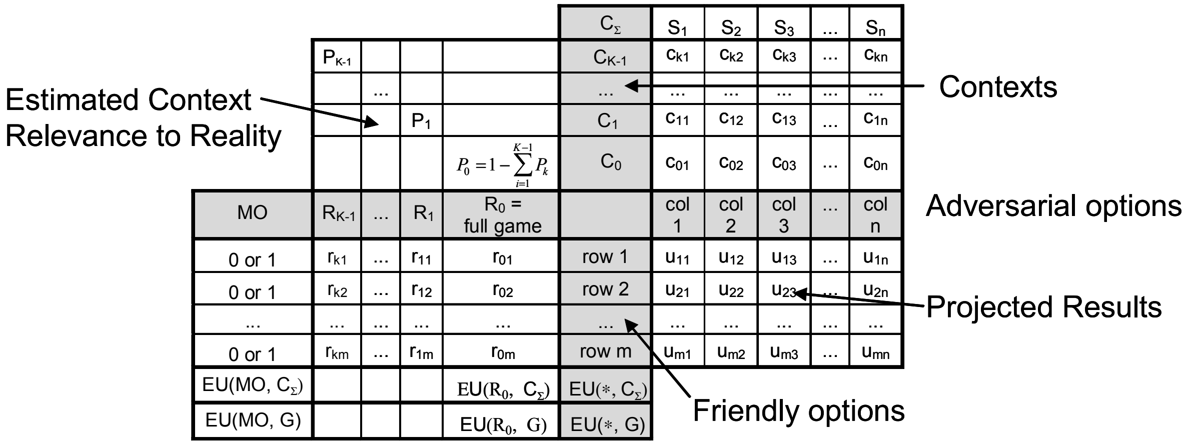
Hypergames

Hypergame theory attempts to bound the uncertainty associated with a decision problem by augmenting the standard game theoretic table of options and situations with contexts and beliefs about evidence. By providing a location to record what might occur so the decision-maker understands how prone to surprise that a decision is. Hypergame normal form is provided in figure 2 and a notional hypergame expect utility plot is provided in figure 3. Hypergames can be used to record the possible adversarial scenarios without forcing consensus. This way the evidence might point the organization to truth.

Figure two shows a very dense representation of *k* contexts, *n* situations/adversary's options, and *m* friendly options; as well as a way to update real time evidence into one's continually refining view of what is going on.

- Contexts are used to assess enemy mindsets, information or predilections that they may or may not have. Contexts can be constructed by military and cultural scholars to inform planners about behavioral differences.
- Situations or situational hypotheses are used to characterize widely varying natural and adversarial capabilities to affect reality.
- Friendly options represent possible own actions

Hypergame theory actually encompasses divergent views about any situation where tradeoffs become apparent. When combined with a current assessment of which contexts are likely to be in play (see the upper left hand corner), an educated estimate of the opponent/nature can be derived as a vector of behaviors. From this we can judge our options. The entries in the lower left side of figure one actually encode our wishes, our expectations, and our fears.



					C_Σ	S_1	S_2	S_3	...	S_n
	P_{K-1}				C_{K-1}	c_{k1}	c_{k2}	c_{k3}	...	c_{kn}
		...			...	...	...	...	...	...
			P_1		C_1	c_{11}	c_{12}	c_{13}	...	c_{1n}
				$P_0 = 1 - \sum_{i=1}^{K-1} P_k$	C_0	c_{01}	c_{02}	c_{03}	...	c_{0n}
MO	R_{K-1}	...	R_1	R_0 = full game		col 1	col 2	col 3	...	col n
0 or 1	r_{k1}	...	r_{11}	r_{01}	row 1	u_{11}	u_{12}	u_{13}	...	u_{1n}
0 or 1	r_{k2}	...	r_{12}	r_{02}	row 2	u_{21}	u_{22}	u_{23}	...	u_{2n}
...	...	...	...	...	...	...	...	...	...	...
0 or 1	r_{km}	...	r_{1m}	r_{0m}	row m	u_{m1}	u_{m2}	u_{m3}	...	u_{mn}
EU(MO, C_Σ)				EU(R_0, C_Σ)	EU(*, C_Σ)					
EU(MO, G)				EU(R_0, G)	EU(*, G)					

Estimated Context
Relevance to Reality
Contexts
Adversarial options
Projected Results
Friendly options

Figure 2. Hypergame Normal Form

Hypergame theory opens up a whole new way of thinking about adversarial problems that is much more sensible to competitive planners. By starting with a concept that all competitors conceptualize the contests differently, allows researchers to explore eleven new dimensions of reasoning.

1. Difference in current situation assessment (closeness to truth).
2. Difference in understanding of projection (what beats what?)
3. Difference in creativity (what tricks can be added)
4. Difference in information (both at the commitment phase and during the operations)
5. Constraints because of time
6. Difference in robustness, resilience of plans
7. Difference in knowledge, expertise
8. Feints, reserves, Denial & Deception
9. Additional phased resources
10. Real time activity assessment
11. Evidence accrual (related to above)

As shown in Figure 3, hypergame expected utility; the planner can see how uncertainty affects the planning model. Increases in uncertainty almost inevitably favor "hedging" plans that involve less risk. So the idea is to gather intelligence that allows edgy, decisive plans while mitigating risk. One can even consider luck in the mix, as seen below.

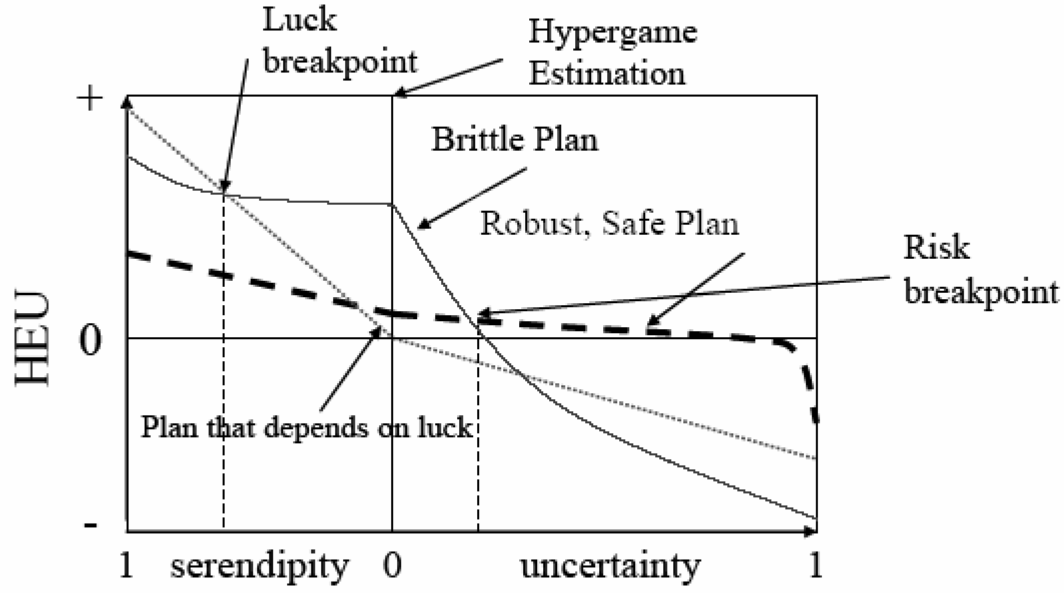


Figure 3. Hypergame Expected Utility (HEU)

Some subtleties are introduced herein. In this plot of the HEU for each of three strategies, the actual reliability or robustness of the plan is plotted. As a first approximation, robust plans are ones with lots of contingencies. Brittle plans are those that depend on a number of parts of the plan being synchronized or occurring in parallel. Straight lines represent those plans that appear to be neither. Note that this allows the planner to account for really bad outcomes in the robust, safe plan (heavy dashed line) – when "everything goes wrong." Additionally, the decision-maker is aware that there should be more concern about maintaining an accurate estimate of uncertainty, for the current "best plan." The risk breakpoint indicates that one need to be reasonably certain about the situational assessment.

A side effect of using hypergames is to decrease the surprise of organizations in general and encourage open discussions about what might happen. Lastly, Samuelson [Samuelson 2006] has noted that hypergames can be nested inside of other hypergames to decrease the scope of the reasoning problem for computational while not giving up the hyper-hypergame perspective, which cobots support. As an example, while planning for a battle the field commander may not consider a large meteor striking his part of the battlefield; but the king can consider the complete annihilation of the nation's forces.

Surprises

The goal of this section is to list many forms of surprise and identify where they might occur in the cobot architecture. Such surprises should have meaning in an unaided setting. If the outcome of aiding a group of humans was to increase the organization's likelihood of surprise; the system would be a failure. Instead, any surprise that is allocated to an architectural component invites design ideas about how to mitigate it and decreases the variability of such surprise. One goal is to list many design ideas in this section.

As John Searle [Searle 1980] has pointed out in his Chinese Room thought experiment, computers can not be intelligent, because mechanistic answers that are based solely on automata theory kinds of syntactic responses such as the Turing test are *behavioristic criteria*, not indicators of *intelligence*.

What does it mean for a model to be surprised? If we take the definition of surprise, it is clear that programs are either never surprised, or always surprised.

1) Since they expect nothing their future projection is not at odds with measured evidence.
2) Since they expect nothing and something happens they are always surprised.

If we take a familiar definition of the computer as a Turing machine then only when the next symbols on the tape are able to be processed is that the machine NOT surprised. Thus if it does halt and it is not in the end condition – it is surprised.

Rather than take a purely academic view of the phenomena of surprise, let's explore what surprise might be and the contexts under which it might occur. We will freely admit that context refers to a type not an enumeration.

If a system halts in an unknown state – it is not surprised – it is broken. We are surprised, if we are new to computers. We are intrigued to debug the system or frustrated, otherwise.

If the system reports that it is confused, then it is not surprised. But to repair itself to an unconfused state or to become unconfused (fixed by an external force), it needs help. This external help could come from other automata (programs) or from humans. Such help could communicate with the confused system and have it resolve its confusion according to premade rules or it can be set into a state by an external force.

In table 1, the type of surprise is tagged (given a short label) in the first column, marked as Expected or Not during design, with the mitigating approaches used by the two kinds of agents – software and human – in the final two columns. This table will also refer to cobot modules and information flows. This table is structured to trace info flows from the least processed (rawest) sensor data in the lower left of figure one up to fully evaluated reports in the upper left. The term "MAU" means that the event "**m**ay **a**ffect **u**ncertainty" in the reasoning process.

Event Type	Expected or Not	Software Agent	Human Supervisor
Sensor data	Expected	Convert to information	Forgotten or “clumped”
	Not	Ignored, MAU	Clumped with noisy, MAU
Information updates	Expected	Convert to evidence, pattern, source for use in model	Forgotten or “clumped”
	Not	Recorded, MAU	Should human clarify? MAU
Situation	Expected	Should be expected with cobot designed for it	Forgotten or “story advances”
	Not	Manageable by next level agent? MAU Report to higher agent	Cope/Continue? MAU Or report “Surprised?”
Plan Results	Expected	Above expected threshold	Forgotten or “story advances”
	Not	Manageable by next level agent? MAU Report to higher agent	Cope/Continue? MAU Or report “Surprised?”
Process Error	Expected	Exception Trap (see below)	Drink Coffee, then:
	Not	Report to higher agent and Restart lower cobot at in-progress position	Change Mission or Restart CoBot
Report to upper	Expected	Tracking within tolerance	Forgotten or “story advances”
	Not	Manageable? MAU Report to higher agent	Cope/Continue? MAU Or report “Failing”
Perceived Situation from lower cobot Report	Expected	Tracking within tolerances	Forgotten or “judged in bounds”
	Not	Can cobot deal with it, or Report to higher agent MAU	Can human ‘clarify’ to cobot Cope/Continue? MAU Or report “Surprised?” or “Failing”
Strange Mission	Not	“Surprised.” Cobot calling protocol should prevent. Report confusion.	“Surprised.” Report confusion.

MAU = may affect uncertainty

Table 1. Not expected versus Surprised

By surveying and comparing the “expected” and the “not expected” events in table one for a single CoBot and its supervising human, the reader can find five actual “Surprise” results. Only one of these is for the CoBot component; when it is mistakenly called to evaluate a mission the cobot does not perform. This should be prevented by the network OS, DARPA’s Cognitive Agent Architecture, COUGAAR. The other four affect higher level surprise. They are: surprising situations (from evidence), surprising plan results (from model), surprising notifications from called cobots, or lastly from receiving a strange mission.

The human and sometimes higher level cobots are responsible for managing the context of the “surprising situation” by accounting for “failing” or “overachieving” situations by changing the mission. This is similar to adjusting goals based on performance to date. Surprising “plan results” occur during planning or plan monitoring. Thus, the model is predicting a sharp change. This is often handled by clarifying the perceived situation which brings the plan projection back into alignment or modifying the model output. If a cobot indicates “surprise” in its report, the calling cobot or human might want to clarify as before or even reinstantiate (restart and initialize) the cobot. Lastly, strange missions (those that are not understandable or wildly out of context, require some validation from above and then often some human creativity.

In a network of collaborators this can be advice, reassignment to another human and cogbot., or real-time problem solving with the specification for a new cobot. But what if the surprise originates from the society of agents becoming confused? This actually can not occur without one of the cobot related “unexpected” events. No cobot *except the top one started by the human* is completely alone. The contexts and projection models are implementations of explicit modeling.

Unaddressed design surprises are listed below, but should be solved by design ideas and Evo Delivery to manage risk. They are in most concrete to most abstract order:

- Unavailable data
- Evidence -> not expected by agent
- Evidence -> expected by agent, hence expected by software designer (type) and user expert (at least one).
- Evidence that is too precise/imprecise
- Evidence that is out of bounds
- Unavailable information
- Unexpected belief(s)
- No applicable hypotheses
- Unknowable things – either via Heisenberg Principle or the Mind of God.
- Model incompleteness
- Unexpected causal relationship(s)
- Unavailable knowledge about process and outcome
- Discontinuities in timing, location, power, order, and intent
- Unexpected results types
- Unexpected results measurement
- Unexpected intent
- Unforeseen strategy(ies)
- Unobservable/unknowable strategy
- Goals unachievable by standard strategies – strategic surprise.
- Unknown context

So there are design challenges to accomplishing cobots at the level of completeness that only humans would need to handle "surprise."

Finding

Hypergames provide an approach to bounding surprise mathematically by referring to unmodeled contexts to conditional evidence collection. They also help the decision-maker to share information and structure upcoming decisions.

Neosymbiosis, using cobots may structure "unexpected" evidence enough that organizations will rarely be surprised in the future [Bennett, Vane 2006]. This bold assertion requires an organizational culture that welcomes dissenting views and is not fixated on technology for its own sake. Previous approaches such as game theory however have penalized such "open" organizations by lowering the expected value of the decision-maker who allows contrarians and pessimist to add their perspectives. Hypergames do not force that result.

Conclusion

Humans, intelligently augmented by systems designed to quickly establish truth and track reality (using software agents) may revolutionize organizational effectiveness.

GDAIS needs to build and test these hypotheses to move beyond a theoretical contribution. In the event that cobots are not able to handle significant aspects of the implied information and contexts required by the test environment, future conclusions will not be supportable. The authors propose to provide some evidence as soon as a new variant of GDAIS's CogBot (a network of cobots) is built to perform in a dynamic, emergent environment. We have several opportunities and government research facilities that may provide the support needed to provide findings in the next symposium or a future AAAI annual meeting.

References

Bennett, M., Vane, R. 2006 Using Hypergames for Deception Planning and Counterdeception Analysis, Defense Intelligence Journal (to be published in upcoming D&D Issue)

Freedy, A. 2006. Private discussions of Amos's work at Perceptronics Solutions, held at GDAIS in Northern Virginia.

Gilb, T. 2005. Competitive Engineering, Elsevier Butterworth-Heinemann

Griffith, D. 2005. Beyond Usability: The New Symbiosis, Ergonomics in Design, 13, 30-31.

Griffith, D., Greitzer, F.L., (in press) Neo-Symbiosis: The Next Stage in the Evolution of Human Information Interaction. International Journal of Cognitive Informatics and Natural Intelligence

Kahneman, D. 2002. Maps of bounded rationality: a perspective on intuitive judgment and choice. Nobel Prize lecture, December 8

Licklider, J.C.R. 1960 Man-computer symbiosis. IRE Transactions on Human Factors in Electronics., HFE ,4-11

Samuelson, D.A. 2006. "The Hyper-Hypergame: Issues in Evidence-Based Evaluation of Social Science," Capital Science Symposium, March 26, 2006

Schum, D. 2005. Thoughts About a Science of Evidence, Studies of Evidence Science, University College London

Searle, J.R. 1980. "Minds, Brains, and Programs," The Behavioral and Brain Sciences, vol.3, pp. 417-24

Vane, R.R., Griffith, D. 2005. "Augmenting Cognition for Adversarial Reasoning", First International Conference for Augmented Cognition, Las Vegas, NV

Vane, R.R., Griffith, D. 2006. Cognitive Automation Solves Many AI-Hard Problems, AAAI Spring Symposium, Stanford, CA

Vane, R.R. 2006. Advances in Hypergame Theory, DTGT Agent Workshop Notes, AAMAS 2006, Hakodate Japan