Exercising Qualitative Control in Autonomous Adaptive Survivable Systems*

Jon Doyle[1] and Michael McGeachie[2]

[1] Department of Computer Science
North Carolina State University
P.O. Box 7535
Raleigh, NC 27605-7535, USA
Jon_Doyle@ncsu.edu
http://www.csc.ncsu.edu/faculty/doyle

[2] Laboratory for Computer Science
Massachusetts Institute of Technology
200 Technology Square
Cambridge, MA 02139, USA
mmcgeach@mit.edu

Abstract. We seek to construct autonomous adaptive survivable systems that use *active trust management* to adapt their own behavior in the face of compromises in the computational environment. Active trust management maintains probabilistic *trust models* that indicate the trustworthiness of different resources for different tasks, and uses these models in rationally adapting allocations of computational resources to tasks. Flexible adaptation of allocations to changing circumstances places great demands on the methods used to represent the utility information needed by rational decision-making mechanisms. This paper explains how to use qualitative preference specifications to exercise effective control over quantitative trust-based resource allocation by facilitating convenient specification and adaptation of the stable foundations of the trust manager's utility judgments.

1 Introduction

The engineering of autonomous systems generates very different demands on representations of decision-making information than those recognized and addressed in traditional decision theory and decision analysis. The traditional approaches mainly focus on manual methods for constructing quantitative utility models that guide rational decision-making systems. Decision theory provides the conceptual properties the utility models must satisfy, and decision analysis provides techniques for eliciting utility information from human informants. Once in place, however, decision making proceeds

* © Copyright 2001, 2002 by Jon Doyle and Michael McGeachie. Version of January 29, 2002 at 16:04. To appear in Proceedings of the Second International Workshop on Self-Adaptive Systems, 2002, under copyright transfer to Springer-Verlag, http://www.springer.de/comp/lncs/.

with the specific, fixed utility model so constructed, leaving all variability of decision to changing probabilistic measures of belief. Such fixed utility models serve poorly in designing autonomous agents, which must have the ability to adapt to face new situations and tasks by changing utility models as well as beliefs. Numeric utility models, of the sort constructed in traditional decision analysis, simply do not provide the conceptual structure needed to facilitate effective adaptation of utility models because mere numerical mappings need not expose the rationales or considerations underlying the assignment of different utility values to different alternatives.

This paper discusses the use of qualitative representations of the comparisons underlying numeric utility measures to provide guidance to autonomous systems. Rather than leave generic and qualitative utility considerations implicit in the decision-analytic process, we expose and represent these considerations directly in order to reason about them and to naturally separate those portions involved in some change from the portions persisting independent of the change. We use the task of guiding the response of autonomous systems to security violations to motivate and illustrate the methods, focusing in particular on the method of active trust management.

2 Active Trust Management

Computational system designers today find themselves in a hostile, even malicious, environment in which no one is safe and nothing is trustworthy. Automated processes scan networks seeking vulnerabilities, to which anyone can succumb anonymously without recognized enemies. Serious adversaries, in turn, can penetrate or defeat any known system; sufficient resources always mean success for the attacker willing to exploit all possible vulnerabilities, including those of the people operating the target systems.

This predicament poses problems for the traditional security idea of constructing a trusted computing base (TCB). Such trusted systems do not exist today, and little suggests they will exist in the future. Absent such a trusted base, however, the natural paranoia of security providers fosters paralysis. Attackers need only create a belief in the possibility that a system has been compromised to defeat it. Given the openness of modern societies and the ever expanding range of technologies, the TCB approach must suffer continuing fragility.

We believe that facing the security predicament without paralysis requires abandoning the all-or-nothing conception of trust underlying traditional TCB conceptions. We approach the problem of providing computational security using the notion of *Active Trust Management* (ATM) [1]. This approach involves three principal notions: construction of a probabilistic trust model, maintenance of the trust model through perpetual analytic monitoring, and rational trust-based resource allocation. We summarize each of these notions in turn.

2.1 Fine-Grained Probabilistic Trust Models

Abandoning all-or-nothing trust concepts requires recognizing that not all compromises affect all tasks, and that not every affected task matters.

Compromises differ in impact. Various computer systems at MIT have on occasion suffered intrusions in which an outsider uses a guessed user password to hijack a host's FTP server. The intruder stores pirated software or images on the server and advertises the location to friends, who then proceed to make copies from the server. Increasing levels of download activity eventually alerts users to the hijacking by making the compromised system slower and slower. This compromise clearly affects the trustworthiness of a host for performing demanding computational tasks. The same compromise, however, does not affect other properties of the host, such as privacy of user email, passwords, or file integrity. In contrast, systems suffering rootkit attacks aimed at gaining root privileges for the intruder need not compromise computational performance, but clearly compromise privacy and file integrity.

Affected tasks differ in importance. One might keenly miss an email service disused because intruders have replaced the normal process with one that copies all traffic to an enemy, but not care a whit about having a SETI@Home screen-saver starved of cycles.

Avoiding paralysis requires judging trustworthiness of resources in terms of the properties relevant to the performance of specific tasks. The trust model thus distinguishes many different properties of resources, some of common interest across tasks, but others of relevance only for certain task types or even task instances. In addition to this fine-grained differentiation between properties, the trust model also employs a fine-grained differentiation between degrees of trustworthiness. Toward this end, the model uses a probabilistic representation of the trustworthiness of each computational resource in the environment with respect to each property of interest.

Our trust models provide detailed decision-theoretic assessments of trustworthiness, suspicion, and related concepts as applied to information systems and their components, including attractiveness of a system as a target, likelihood of being attacked, likelihood of being compromised by an attack, riskiness of use of the system, importance or criticality of the system for different purposes, etc.

2.2 Perpetual Analytic Monitoring

Perpetual analytic monitoring seeks to keep the trust model current, reflecting the best estimates in light of observations. Such monitoring employs numerous different data sources, including sensors attached to different resources and diagnostic data streams generated by self-instrumenting task processes. We use the MAITA monitoring infrastructure [2] to provide the underlying monitoring framework and architecture. This architecture supports a network of distributed monitoring processes that analyze and correlate the sensor and diagnostic data streams, generating alerts or more refined data streams for further correlation or diagnostic stages.

Constructing and maintaining the trust model requires making estimates of the likelihood that a resource has been compromised in a particular way. Perpetual analytic monitoring thus must differentiate between different compromise events. To do this, MAITA employs abstract event descriptions called *trend templates* [3, 4], each of which characterizes a pattern of activities over several temporal intervals and in terms of several data streams, and a suite of mechanisms for matching trend templates to data in performing event recognition. The language for expressing trend template includes means for characterizing *landmark times* that represent possibly abstract divisions between

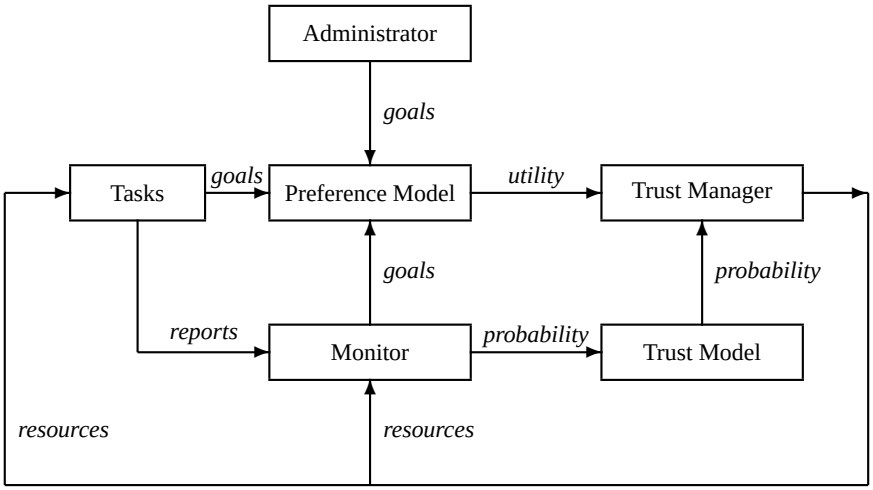

Fig. 1. The active trust management control loop.

different stages of events, algebraic relations between temporal intervals, and behavioral patterns of data values, such as constant, increasing or decreasing, and oscillating signals of different magnitudes, shapes, and frequencies. Matching processes employ various statistical criteria. The trust models themselves involve Bayesian network structures.

2.3 Rational Trust-Based Resource Allocation

The sensible response to possible or recognized compromises consists of making do with what one has, with doing the best one can with degraded resources. Active trust management interprets these common-sensical prescriptions as choosing resource allocations that maximize expected utility of task-suite performance. The estimates of expected utilities involved in these decisions depends on the probabilities captured in the trust model and maintained by the self-monitoring capabilities. The expected utility estimates also depend on a model of the utility of different task performance properties. This utility model reflects goals set by the processes charged with accomplishing primary system tasks, goals set by the system monitor, and goals set by external system administrators. The task and monitoring processes set and revise goals reflecting ongoing subactivities and changing priorities among these subactivities. One can expect that, in contrast to these internal autonomous changes, external guidance from administrators or users occurs only infrequently and episodically. Figure 1 depicts the flow of information and resources characteristic of active trust management.

3 Exercising Control in Active Trust Management

Control of active trust management takes several forms.

– An administrator or system process can establish and change the structure of the monitoring network in order to monitor different conditions. This structure reflects knowledge about relationships between different signals, such as independence or correlations, that exist independent of specific security concerns. It also reflects, however, the goals or concerns of a security manager, with specialized processes added to track temporary or specific concerns.
– An administrator or system process can adjust the models that guide alerting decisions. These models encode information about the likelihood different recipients will want to see different classes of alerts, at different times, via different communications paths, and in different forms. The models also encode the value of alerts to the sender and recipients in the different cases. In simple cases, alerting decision models consist of simple thresholds one can raise or lower, but more useful alerting models provide a more expressive representation of the preferences or utilities guiding the decisions.
– An administrator or system process can indicate the utility model guiding resource allocation decisions. This model, like the more sophisticated alerting models, involves expression of complex dependence of the utility of resource allocations on properties of task performance.

Putting these control mechanisms together, we find that exercising control over active trust management involves specification of goals and utility information. In the following, we consider means for specifying such goals and utility information in a coherent and coordinated way. We simplify and shorten the presentation by focusing on specification of resource allocation utilities. We take a similar approach to specification of alerting utilities.

3.1 Representational Concerns About Utilities

Though many computational decision-making methods rely on numeric utility functions, such functions provide poor representations for knowledge about task importance. Utility functions, of course, merely serve as numeric representations for orderings of relative preferability or desirability. They add irrelevant details to the underlying ordering, in the sense that many different numeric utility functions can represent the same underlying ordering.

In fact, utility functions and their underlying ordering really do not distinguish between essential and inessential aspects of preferability. Common judgments of preferability involve both fundamental preference relations and specific tradeoffs. One might, for example, judge the security of login services more important than the speed performance of those services. One might persist in that judgment even as one changes opinions on just how much more important login security is than login performance.

This confusion of preferential essence and accident impedes convenient specification of changes. One might start thinking performance more important than security for some purpose, and choose some specific tradeoff ratio. Later experience might provide evidence that this preference was wrong, that security is instead more important than performance. The natural inclination is to switch the underlying preference and figure out later the new tradeoff value. However, if all one has is a numerical utility function

encoding the former tradeoffs, one lacks any simple means for adjusting it to reflect the switch in underlying preferences.

Worse still, an autonomous system seeking to maintain its own security can expect to encounter unforeseen circumstances and to occasionally modify its goals. Both of these events may call for modifying the utility function in even more fundamental ways than merely switching an existing preference, for they may call for adding new goals or conditions of interest, adopting new preferences, perhaps by means of inference from general principles set up to guide the system in facing novel circumstances.

3.2 Qualitative Preferential Information

To provide a framework for effective autonomous action in resource allocation, we step back from direct provision of numerical utility functions and augment utility functions with qualitative representations of preference information. Here we transcend the familiar non-numerical orders among individual outcomes underlying utility functions to express or capture abstract and generic preferences among qualities or properties of outcome classes. These qualitative expressions provide constraints on the orderings of individual outcomes, and provide a natural basis for reasoned construction of decision models appropriate to novel circumstances or types of decisions.

These qualitative representations of underlying preference structures augment rather than displace numeric utility functions, which remain important in calculations intended to optimize expected utility. We connect the two notions by providing mechanisms to automatically construct numeric utility functions from sets of qualitative specifications. This permits a human administrator or system process to exercise control through generic, qualitative expressions and the automated resource allocator to exercise control through concrete, numeric expressions.

4 Illustrating the Concepts

To help convey these abstract concepts, we illustrate the approach using a simplified form of the qualitative language under development and an imaginary but plausible application setting in which the trust manager operates to serve a company that performs computer animation for films. For simplicity of exposition, we start by examining how a human administrator can exercise control along the lines outlined in the preceding, and then consider means by which system processes might exercise similar control.

The company—let us call it Acme Animation Associates—does the usual things with its computers, such as logging in, handling email, maintaining marketing, operational, and financial databases, and running a web server. Acme's core operation, however, consists of running animation processes that render films frame by frame.

Acme's system administrator, who we will call Al Greenspan, is charged with providing the underlying guidance to Acme's ATM system. To do this, he starts by identifying the fundamental goals in Acme's local setup. These include some standard goals, such as performance and security for individual services and the overall system, and may include addition goals appropriate to the special concerns of the local environment, in Acme's case, having to do with its animation activities. For simplicity, we

Service	*Performance goal*	*Security goal*
login	P.login	S.login
mail	P.mail	S.mail
web	P.web	S.web
db	P.db	S.db
animation	P.animation	S.animation
seti	P.seti	S.seti

Fig. 2. Acme's ATM goals

Service	*Service-specific preferences*
login	S.login $>$ P.login
mail	S.mail $>$ P.mail
web	P.web $>$ S.web
db	S.db $>$ P.db
animation	P.animation $>$ S.animation
seti	P.seti $>$ S.seti

Fig. 3. Pure service preferences

suppose Greenspan starts by identifying only the performance and security goals for each of the standard services, to which, as an astronomy buff, he adds SETI@Home. This produces the goals displayed in Figure 2. Here one can think of "performance" as meaning something like "good performance". For the purpose of this illustration, we will only consider the coarse distinction between good and not-good (or bad) performance. More refined models can employ more refined categories of performance levels or characteristics.

With the system goals specified, and presumably with monitoring processes installed to observe and quantify these properties, Greenspan next specifies underlying preferences among these qualities. The first set of preferences relate only goals for the same service, as indicated in Figure 3. Greenspan also specifies some preferences that relate goals for different services, as given in Figure 4. Greenspan leaves a number of goals unrelated, as their relative ranking does not seem to matter to him as long as all get done.

With this initial specification of goals and preferences, Greenspan tells the ATM system to making his guidance effective by constructing a utility function over resource allocations compatible with the generic preferences he has specified. This step is necessary because we distinguish preferences among qualities, as expressed in the preceding figures, and which we can more correctly notate as $>_q$, from preferences between individual resource allocations, which we can notate as $>_a$. The preferences among qualities do not refer to specific resource allocations, but only to properties of the allocations, such as whether the allocation in question provides good database performance or good web-server security. Preferences among allocations instead compare specific allocations of resources to tasks. If we suppose for simplicity that such allocations concern

Cross-service preferences
S.login >S.web
P.web > P.seti
P.db > P.seti
P.animation > P.seti
S.web >P.seti
S.db >P.seti
S.animation >P.seti

Fig. 4. Mixed service preferences

$$\begin{vmatrix} \text{P.web} = 10 \\ \text{S.web} = 3 \\ \text{P.db} = 15 \\ \vdots \end{vmatrix} >_a \begin{vmatrix} \text{P.web} = 12 \\ \text{S.web} = 80 \\ \text{P.db} = 1 \\ \vdots \end{vmatrix}$$

Fig. 5. A comparison between specific allocations

only allocation of units of CPU cycles to different tasks—hardly a desirable supposition in realistic situations—we might write one such comparison as in Figure 5. The ATM utility-construction task, then, consists of constructing a numeric utility function over individual allocations that conforms to the qualitative preferences, or roughly speaking, a function U such that $U(x) >_a U(y)$ if $x >_q y$. The ATM system constructs a suitable utility function and proceeds to allocate system resources.

The day comes, however, when trouble strikes and Greenspan is in the hot seat. Late one Friday (of course), the payroll manager confronts Greenspan because the production of payroll checks is not finishing as expected. On her heels the sales manager stomps in to inform Greenspan that someone has defaced Acme's web site with a crude insult to Acme's prime customer. Greenspan looks into the payroll problem first and determines that the ATM system is starving the database of cycles, giving preference, as usual, to the core animation processes. He promises to conduct a longer investigation to find out how the web intrusion occurred, but does notice he had specified that web performance was preferable to web security.

With these quick determinations, Greenspan modifies his guidance to the ATM system. For lack of a more specific solution to the defacement problem, Greenspan reverses his earlier guidance and tells the ATM system that good web server security is preferable to good web server performance. To address the payroll problem, Greenspan introduces some new goals and preferences. The new goals refine the existing goals for database and animation process performance by introducing goals specific to paydays. He then adds preferences that indicate database performance on paydays preferable to animation performance on paydays. We depict these revised preferences in Figure 6. Greenspan instructs the ATM system to reconstruct and install an updated allocation utility function.

Revised preferences
S.web $>$ P.web P.db.payday $>$ P.animation.payday

Fig. 6. Changed and added service preferences

To summarize, Greenspan expressed the principles underlying the desired ATM performance in fairly natural terms by stating and relating various qualitative goals for different services and at different levels of specificity. He was later able to state changes to these principles in natural ways. The ATM system, in turn, took care of translating these principles into quantitative guidance for the decision making mechanisms.

Autonomous responses to novel circumstances require some knowledge to guide the responses. The scenario sketched assumes a system unaware of the sorts of situations addressed by Greenspan's corrections. A more knowledgeable system might well prove capable of instituting some corrections on its own. Although it seems unreasonable to assume a general-purpose ATM system knows about customers and insults, having the system know about task deadlines seems perfectly reasonable. In the example above, the payroll task might have instead monitored its own progress, projected a failure to meet its deadline with the resources it was obtaining, and itself generated the temporary high priority for the payroll task relative to the core animation process without requiring any intervention by Greenspan to effect this correction.

Indeed, the flexibility of the architecture depicted in Figure 1 means that the an autonomous system need not make prior provision for all deadline dangers, but instead can choose to address some categories of deadline tasks by constructing responses only as impending deadlines draw uncomfortably near. In comparison with fixed schemes for deadline-dependent utility functions, such as those explored by Haddawy and Hanks [5], on-the-fly crisis management permits exploiting special characteristics of the specific circumstances faced at the time. In some cases, these special characteristics can provide value that outweighs the usual disadvantages of crisis-driven response. Indeed, in novel circumstances, prior provision proves impractical, and one must of necessity rely on crisis-driven responses.

5 Formal Tools

We have not yet mechanized all the capabilities alluded to in the illustration just presented. In the following, we describe two important formal techniques, namely a logic of generic preference capable of expressing both qualitative goals and preferences among them, and a utility construction method to make such guidance effective.

5.1 Specifying Preferences and Goals

The standard decision-theoretic framework for rational choice under uncertainty (see, for example, [6]) involves

- A set $\mathcal{A} = \{A_1, A_2, \ldots\}$ of *actions*,
- A set $\mathcal{S}$ of states or *outcomes* that can result from actions and in which the agent takes action,
- A probability distribution $Pr_A : \mathcal{S} \rightarrow \mathbb{R}$ for each action $A \in \mathcal{A}$ that assigns to states the agent's degree of belief that the states result from A,
- A *weak preference* order $\precsim_{\mathcal{S}}$ over states that forms a complete preorder, that is, a complete reflexive and transitive order over alternatives, which compares desirability of outcomes, where $S \precsim_{\mathcal{S}} S'$ means that the agent finds S' at least as desirable as S,
- A numerical *utility* function $u : \mathcal{S} \rightarrow \mathbb{R}$ that represent $\precsim_{\mathcal{S}}$ in the sense that $A \precsim_{\mathcal{S}} B$ iff $u(A) \leq u(B)$,
- An *expected utility* function $\hat{U} : \mathcal{S} \rightarrow \mathbb{R}$ such that $\hat{U}(A) = \sum Pr_A(S)u(S)$,
- A weak preference order $\precsim_{\mathcal{A}}$ of over $\mathcal{A}$, defined by requiring that $A \precsim_{\mathcal{A}} B$ iff $\hat{U}(A) \leq \hat{U}(B)$, so that the most preferred action in a set is one that maximizes expected utility.

One defines the two additional relations of *indifference* among alternatives, written $A \sim B$, so that $A \precsim B$ and $B \precsim A$, meaning that the agent finds whatever differences exist between the alternatives leave them equally desirable, and *strict preference*, written $A \prec B$, so that $A \precsim B$ but $B \not\precsim A$, meaning the agent finds B more desirable than A.

Our specifications of preferences and goals starts with a comparison relation between propositions, which represent generic outcome classes rather than the comparisons of individual outcomes of the standard formulation. We introduce notation for propositional preferences and goals by writing, when p and q denote sets of outcomes, $p \trianglerighteq q$ to mean the proposition p is weakly preferred to the proposition q, and $\trianglerighteq (p)$ to mean that p is a goal, which we interpret as shorthand for $p \trianglerighteq \bar{p}$, where $\bar{p} = \Omega \setminus p$ denotes the complement of p.

Wellman and Doyle [7] showed that we cannot usefully interpret propositional preferences of this kind in terms of simple lifting of preferences over outcomes. They instead proposed interpreting propositional preference as preference *ceteris paribus*, so that $\trianglerighteq (p)$ means that the agent prefers outcomes in p to outcomes in $\bar{p}$, other things equal. In particular, they interpreted the *ceteris paribus* clause in terms of a multiattribute representation of outcomes, so that propositional preference compares outcomes that vary on the attribute of interest but hold all other attributes constant.

Doyle, Shoham, and Wellman [8] then extended this to general comparisons $p \trianglerighteq q$ and proved the soundness of various principles for inferring propositional preferences from others. These sound inference principles included cases of dominance or sure-thing reasoning, such as inferring $p \trianglerighteq q$ from $pr \trianglerighteq qr$ and $p\bar{r} \trianglerighteq q\bar{r}$; goal inference, such as inferring $\trianglerighteq (p)$ from $\trianglerighteq (q)$ and $p \trianglerighteq q$; and goal combination, such as inferring $\trianglerighteq (p \wedge q)$ and $\trianglerighteq (p \vee q)$ from $\trianglerighteq (p)$ and $\trianglerighteq (q)$. Some of these proofs held only for propositions expressed in certain syntactic forms. Doyle and Wellman [9] later developed an alternative semantic basis for the logic that provides the results without the syntactic restrictions.

Other representations, logics, and semantics for propositional preferences have been investigated by a variety of authors, including Pearl and Tan [10, 11], Boutilier [12, 13], Bacchus and Grove [14, 15], and Shoham [16, 17]. Some of these focus on the notion

of *utility independence* instead of individual propositional preference comparisons (see also [18]). Each of these systems has its own advantages and disadvantages. The logic of preference *ceteris paribus* supports significant inferential capabilities, but seems overly strong for some purposes. In particular, it does not provide a means for expressing preferences over more restrictive propositions that reverse preferences (form an exception to) preferences expressed over more general propositions. This poses problems for formalizing the revision of preferences, depicted in the illustration earlier, so that database performance dominates animation performance on paydays but animation performance dominates database performance otherwise. Addressing such needs may simply call for somewhat different formulation of goals, but might also require changing the underlying logic of propositional preferences. Such a change can have disadvantages, however. For example, logics based on conditional logics make expression of exceptional subcases easier, but in turn support almost no inferences. Solving these problems requires further research on qualitative preference, which continues as part of a larger investigation of qualitative decision theory [19].

5.2 Constructing Utility Functions

Our ongoing work [20] addresses the task of constructing a utility function over outcomes compatible with a set of qualitative preferences. We construct a utility function essentially by starting with presumptions of utility independence between all qualities and then using explicit statements of generic preference ceteris paribus to identify clusters of utility-dependent qualities. The construction exploits the presumptive utility independence to simplify the structure of the constructed utility function as much as possible, making the construction of low complexity in many cases. The method employs an intermediate representation that transforms individual preferential comparisons into small sets of relations between fundamental propositional classes, and applies various graph-theoretic algorithms to separate relations between fundamental classes into utility-dependent clusters. Further graph methods yield subutility functions over these clusters, and an additive scaling of the subutility functions yields a utility function compatible with the qualitative specification, as desired.

6 Conclusion

Autonomous adaptive survivable systems use active trust management, based on graded knowledge about the trustworthiness of resources for different purposes, to accomplish as much as possible with possibly degraded resources. While this performance may fall short of system needs in some circumstances, it represents the best one can do. Obtaining these results, of course, depends on one adequately characterizing the value of different computational outcomes in relation to each other. Systems operating with a changing population of circumstances and types of tasks make it imperative that human and automated controllers find it as simple as possible to specify and revise these values. The arguments and illustration presented in this paper indicate that qualitative propositional preferences provide a natural and effective means for exercising effective control

over active trust management. Realizing the promise of this approach requires considerable further work, but the history of progress in effectively representing probabilistic information offers considerable hope that similar progress will obtain in bringing qualitative preference techniques into widespread use.

Acknowledgments

We thank the anonymous referee for helpful suggestions, and thank Howard Shrobe, Robert Laddaga, Peter Szolovits, and William Long for many useful discussions. This work was supported by the Defense Advanced Research Projects Agency of the United States of America under contract F30602-99-1-0509. Mike McGeachie is supported in part by a training grant from the National Library of Medicine, and a grant from the Pfizer corporation.

References

1. Shrobe, H., Doyle, J.: Active trust management for autonomous adaptive survivable systems. In Robertson, P., Shrobe, H., Laddaga, R., eds.: Self-Adaptive Software. Lecture Notes in Computer Science. Springer Verlag, Berlin (2001) 40–49 Revised papers from the First International Workshop on Self-Adaptive Software (IWSAS 2000).
2. Doyle, J., Kohane, I., Long, W., Shrobe, H., Szolovits, P.: Agile monitoring for cyber defense. In: Proceedings of the Second DARPA Information Security Conference and Exhibition (DISCEX-II), IEEE, IEEE Computer Society (2001)
3. Haimowitz, I.J., Kohane, I.S.: Automated trend detection with alternate temporal hypotheses. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Chambery, France (1993) 146–151
4. Doyle, J., Kohane, I., Long, W., Shrobe, H., Szolovits, P.: Event recognition beyond signature and anomaly. In: Proceedings of the 2001 IEEE SMC Workshop on Information Assurance and Security, United States Military Academy, West Point, New York. (2001) 17–23
5. Haddawy, P., Hanks, S.: Utility models for goal-directed decision-theoretic planners. Computational Intelligence **14** (1998) 392–429
6. Savage, L.J.: The Foundations of Statistics. second edn. Dover Publications, New York (1972)
7. Wellman, M.P., Doyle, J.: Preferential semantics for goals. In: National Conference on Artificial Intelligence. (1991) 698–703
8. Doyle, J., Shoham, Y., Wellman, M.P.: A logic of relative desire (preliminary report). In Ras, Z.W., Zemankova, M., eds.: Methodologies for Intelligent Systems, 6. Volume 542 of Lecture Notes in Artificial Intelligence., Berlin, Springer-Verlag (1991) 16–31
9. Doyle, J., Wellman, M.P.: Representing preferences as *ceteris paribus* comparatives. In Hanks, S., Russell, S., Wellman, M.P., eds.: Proceedings of the AAAI Spring Symposium on Decision-Theoretic Planning. (1994)
10. Tan, S.W., Pearl, J.: Specification and evaluation of preferences for planning under uncertainty. In Doyle, J., Sandewall, E., Torasso, P., eds.: KR94, San Francisco, CA, Morgan Kaufmann (1994)
11. Tan, S.W., Pearl, J.: Qualitative decision theory. In: AAAI94, Menlo Park, CA, AAAI Press (1994)

12. Boutilier, C.: Toward a logic for qualitative decision theory. In Doyle, J., Sandewall, E., Torasso, P., eds.: Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference (KR'94), San Francisco, Morgan Kaufmann (1994)
13. Boutilier, C., Brafman, R.I., Hoos, H.H., Poole, D.: Reasoning with conditional ceteris paribus preference statements. In: Proceedings of Uncertainty in Artificial Intelligence 1999 (UAI-99). (1999)
14. Bacchus, F., Grove, A.: Graphical models for preference and utility. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann (1995) 3–19
15. Bacchus, F., Grove, A.: Utility independence in a qualitative decision theory. In: Proceedings of the Fifth International Conference on Knowledge Representation and Reasoning, Morgan Kaufmann (1996) 542–552
16. Shoham, Y.: Conditional utility, utility independence, and utility networks. In Geiger, D., Shenoy, P.P., eds.: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, San Francisco, California, Morgan Kaufmann (1997) 429–436
17. Shoham, Y.: A symmetric view of probabilities and utilities. In Pollack, M.E., ed.: Proceedings of IJCAI-97, San Francisco, California, Morgan Kaufmann (1997) 1324–1329
18. Wellman, M.P., Doyle, J.: Modular utility representation for decision-theoretic planning. In: Proceedings of the First International Conference on AI Planning Systems. (1992)
19. Doyle, J., Thomason, R.H.: Background to qualitative decision theory. AI Magazine **20** (1999) 55–68
20. McGeachie, M.: Utility functions for ceteris paribus preferences. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (2002) In preparation.