REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-98-
0395

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
25 Aug 97

3. REPORT TYPE AND DATES COVERED
FINAL TECH RPT, 01 JAN 94 TO 30 JUN 97

4. TITLE AND SUBTITLE
Strong Autonomy for Physical Domains

5. FUNDING NUMBERS
F49620-94-1-0118

6. AUTHOR(S)
Drs. Nils Nilsson, Pat Langley, Daniel Shapiro

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Robotics Laboratory, Computer Science Dept
Stanford University
Stanford CA 94305

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
110 Duncan Avenue Suite B115
Bolling AFB DC 20332-8050

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited Distribution

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A prototype of a strongly autonomous agent has been implemented. This prototype selects its own objectives and its own values. The agent can then calculate expected values, choose courses of action, and measure received reward.

14. SUBJECT TERMS
autonomy, prototype, artificial intelligence

15. NUMBER OF PAGES
21

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Uncalssified	UL

Strong Autonomy for Physical Domains

Final Report

AFOSR Grant No. F49620-94-1-0118

Nils Nilsson, P.I.

Pat Langley

Daniel Shapiro

Prepared by Daniel Shapiro

8/25/97

1. Introduction

This is the final report for our research study of learning in physical domains. While the work began with a focus on computational models of human learning using Langley's Icarus architecture, it evolved into a somewhat wider inquiry concerning the architectural requirements of strong agent autonomy. This introduction explains the motivations behind that change.

Our initial work focused on human learning. We examined questions of selective sensing, automaticity, and adaptive response, drawing examples from a variety of simple, simulated physical domains: pole balancing, truck steering, and piloting a plane. This work resulted in several published papers (see the Appendix to this report) and attracted a graduate student, Daniel Shapiro, who expressed an interest in employing Icarus to investigate the topics of task selection and abandonment. This presented an opportunity to pursue research in an important and relatively unexplored area of adaptive response.

Our tool set changed as we investigated task selection and abandonment. In particular, Dan brought a Decision Theoretic approach which applied a more normative/rational shading to the descriptive models we had developed. A second consequence emerged from the work itself; we learned that artificial agents can only abandon their tasks by reference to some underlying criterial structure. That structure can also act as a motivator for selecting tasks, and indeed, for grounding a more powerful sense of "strong autonomy" in which the agent acts to service its own interests (and perhaps simultaneously to service ours). Phrased in this way, the concept of strong autonomy raises intriguing issues concerning the meaning of independence, its practical value and its supporting technology. The topic should be of interest to the believable agents community, AI engineers, cognitive scientists, and agencies with challenging agent/robot applications such as the Air Force and NASA.

1.1 Guide to Reading

This content of this report is roughly evenly divided between a discussion of strong autonomy (Sections 2-9), and the Appendix which documents aspects of our work in published (and submitted) papers. We have organized this report as annotated briefing booklet with sections covering the definition of strong autonomy (2), its interest (3), examples (4), the dual roles of such an agent (5), the underlying technical issues (6), a summary of the work we have performed to date (7), our conclusions (8), and our future research agenda (9). We welcome any and all comments concerning this research.

What is Strong Autonomy?

- *Strong Autonomy is the ability to choose.*
  - to act in service of internal objectives
  - to select objectives that service own values
  - to determine what to value in the world

→ **A strongly autonomous agent possesses meaningful independence.**

→ **People are strongly autonomous. To date, machines are not.**

2. What is Strong Autonomy?

We coin the term "strong autonomy" to express the fundamental ability to choose. While this definition is colloquial and highly metaphorical, it names a property we easily recognize in humans; we possess final authority over our own choice in action. For example, we respond to, but are not subservient to the dictates of others. We choose what to do, why, when and how do it. We possess motivations which drive our long and short term behavior, and we ultimately judge ourselves against that personal standard. In metaphor, we are our own playwright, actor and audience (or if you prefer, kingpin, felon and judge). Our behavior is self-directed at core.

As an illustration, consider a military officer. Despite the fact that this person has granted others a great deal of authority over his/her actions, he remains in the position of choice; in the end, he will either obey or refuse orders. There will be consequences either way (potentially severe) but the officer remains the final arbiter of his behavior (and legally responsible for it).

In contrast, agent technology provides examples of what strong autonomy is not. Consider a thermostat which is not a particularly interesting agent *because* its autonomy is limited; it responds in fixed ways to a narrowly conceived environment. An industrial robot deals with a wider set of circumstances but it is generally frozen in purpose, tasked by a human to turn bolt #23. "Autonomous" air, land, sea and space faring robots have some authority to select sub-tasks, but within a goal and priority structure set by people. In summary, no current artificial agent possesses strong autonomy.

In order to move from a metaphorical to a more concrete definition of strong autonomy we introduce a rational action vocabulary: we define an agent as an algorithm which inputs perceptions of an environment and outputs actions in service of certain tasks. We define a rational agent as one that seeks to maximize some measure of received reward through the pursuit of explicit objectives. Given this context we ask, *What agent abilities enable a meaningful sense of choice?* We identify three:

(1) The agent should act in pursuit of objectives. This implies the existence of plans and expectations.

(2) The agent should pursue objectives in service of an agent-held sense of value. We envision encoding values (or more florridly, desires) as an explicit function of state.

(3) The agent's values should not be fixed, since static bedrock will freeze its purpose by design. Instead, the agent's values must evolve through interaction with its world.

We take these three abilities as an operational definition of strong autonomy.

Why Study Strong Autonomy?

- Research addresses deep questions:
  - ***Where do goals come from?***
  - ***Where do values come from?***

- A strongly autonomous agent will be a better tool when the right objective is hard to predict.

3. Why Study Strong Autonomy?

The concept of strong autonomy has obvious appeal in that it borders on philosophical questions concerning intent and free-will. This is not our purpose. Long before such lofty distinctions are captured in a device made of software and steel, we will address other, more detailed issues of wide practical and theoretical interest.

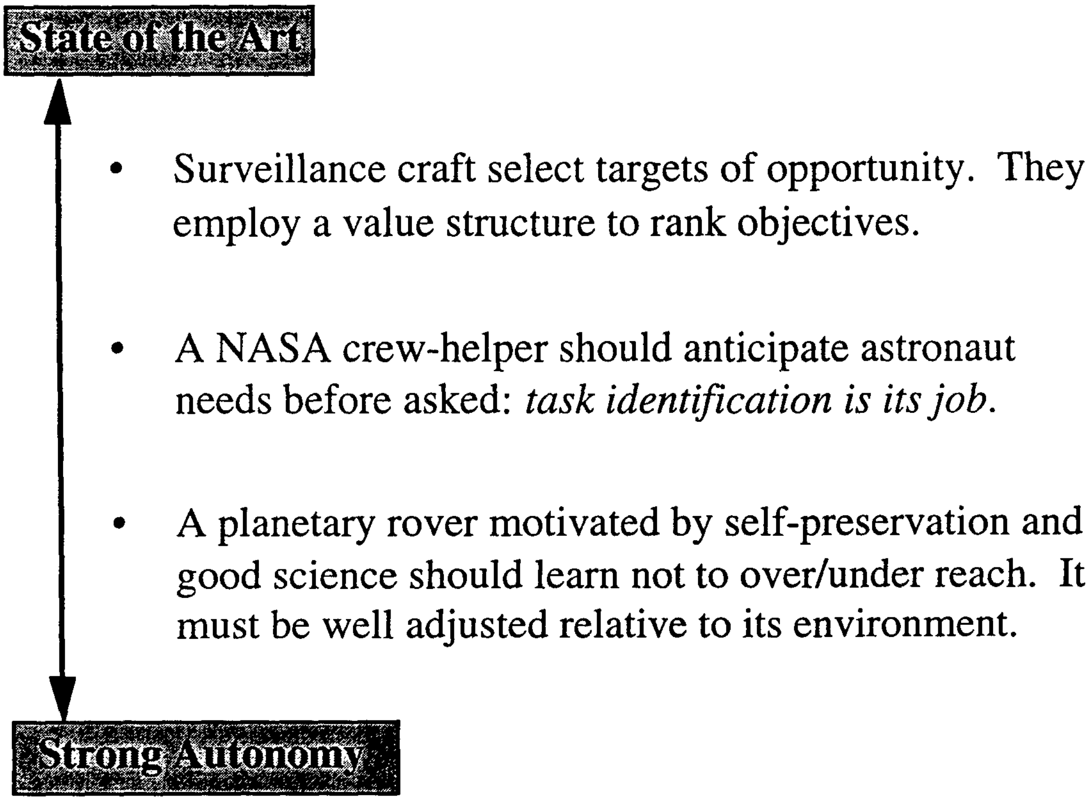
The question, *Where do goals come from?* has lurked in the background of AI for many years. Thirty years of research on planning systems presumes goals are the input, while a few recent efforts in Decision Theoretic AI treat value as the more fundamental construct [Kushmeric]. There is some work on plan execution to maximize utility, and a literature on optimal scheduling. The result is that we can treat value functions (which rank situations by relative preference) as a plausible source for goals, but the approach is subject to practical limitations.

The question, *Where do values come from?* is entirely unaddressed. Since Decision Theory and Economics treat values as predefined, they are the object, not product of modeling. While Operations Research provides a wealth of maximization techniques, all are subject to the given objective. It is tempting to explain this endemic bias as a consequence of role; because people employ technology to solve problems, people own the problem definitions. Thus, selecting the objective function becomes a necessary part of the modeler's art. Asking *Where do values come?* steps outside this frame, suggesting it is a novel and correspondingly interesting research question.

From an engineering perspective it might be hard to convince people that a strongly autonomous agent is a desirable tool. We offer two counter-arguments. The first is that agent autonomy historically increases with time. Machines automated repetitive tasks and software with branch points automated conditional behavior. Programming agents via goals automates the selection of means, and programming by initializing value structures automates the selection of specific objectives. It will be viewed as a good in its time. The second argument appeals directly to good engineering practice; since agent designers are removed in both time and place from the agents which will act in their stead, the agent will sometimes be in a better position to select the appropriate task. The rationale for such autonomy increases with the scope of the agent's concerns, the uncertainty of the environment model, and the inaccessibility of human controllers. We list a range of such applications in the following section.

In summary, research on strong autonomy is interesting on theoretical and practical grounds to multiple communities; AI planning, Decision Theoretic AI, Cognitive Science (for suggestions as to the plausible source of goals and values), and the recent "Agents" group (who seek to develop believable agents with human-like attributes for user interface and entertainment purposes). In the long run, suggestions about the source of intent will benefit Common Sense Philosophy.

Example Agents



State of the Art

- Surveillance craft select targets of opportunity. They employ a value structure to rank objectives.

- A NASA crew-helper should anticipate astronaut needs before asked: *task identification is its job.*

- A planetary rover motivated by self-preservation and good science should learn not to over/under reach. It must be well adjusted relative to its environment.

Strong Autonomy

4. Example Agents

The examples at left should clarify the potential benefits of strong autonomy. We list them in increasing order of required autonomy, novelty, or difficulty. The underlying spectrum concerns *task uncertainty*, which we define as our lack of confidence that a design time specification of the agent's objective will be appropriate at execution time. Task uncertainty increases as the scope of agent's function grows, and as design time knowledge of the agent's run time environment degrades. High task uncertainty requires greater agent autonomy.

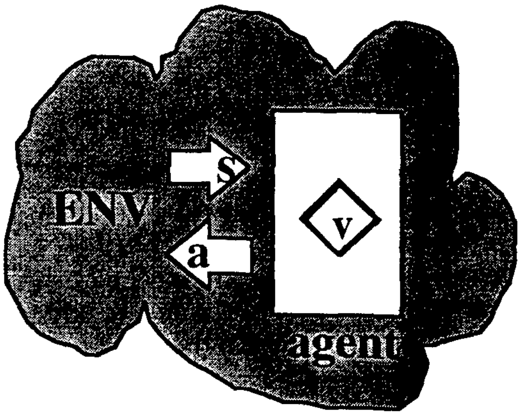
Consider an autonomous surveillance craft whose purpose is to observe interesting objects in remote, hostile areas. Since the designer cannot predict the activities or locations of the objects in advance, and communication is disallowed, the agent needs the ability to prioritize its observations at run time. The necessary criteria can be phrased as metrics of interest or as explicit condition-action cues. In either case the task uncertainty of the application demands a degree of autonomy that is just possible to produce today.

NASA is interested in building a general purpose crew-helper for extra-vehicular activity. Since EVA work is high risk and astronaut time is precious, any transfer of labor to a robot (such as gofer, inspection, and third-arm tasks) is well motivated. In fact, crew time is *so* precious that a pure command-response architecture ill-advised, as is a closely supervised principle-agent interaction. The crew-helper should respond to commands, anticipate some needs before it is asked, and select productive tasks when free. The robot's job is to provide this level of autonomy.

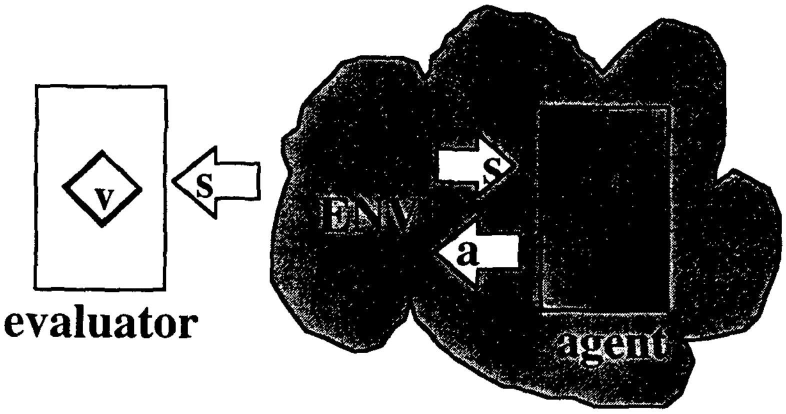
Looking further into the future, planetary rovers will require both a wide-ranging competence and the ability to operate in extremely uncertain environments. The Europa explorer (considered for 2035) will examine the geology and possible biology in the ocean under Europa's ice, which is an archetype for an unknown and unpredictable environment. This robot will face dangers and opportunities not modeled at design time, and it will remain out of communication with earth for extended periods of time. At the same time, its mission is very broad. The explorer clearly needs an internal representation of its abstract interests, and a broad authority to identify and trade-off relevant tasks. It must also guard against wasting resources (time, energy) conceiving of, or pursuing implausible objectives. For survivability and efficiency, the explorer must learn to scale its interests to its world. It requires strong autonomy to address extreme task uncertainty.

Two Roles of a Strongly Autonomous Agent

- *Agent as principle:* purpose is to deliver value to itself



- *Agent as tool:* purpose is to provide value to a user



5. Two Roles of a Strongly Autonomous Agent

We can clarify the previous discussion by recognizing that a strongly autonomous agent can fulfill *two* distinct roles. In the first, the agent acts as a principle (devoid of a human user) whose purpose is purely to service its own set of values. Such a creature can only be evaluated in its own terms by asking, *Does the agent's behavior makes it "happy"*? That is, does the agent acquire value for itself by acting on its world? Note that no psychological interpretation is implied; we are measuring agent performance along an agent-held numerical scale.

The second role casts the agent purely as a human-wielded tool. To crystallize this perspective, imagine a user who can observe the robot interacting with its environment but is otherwise passive. This user owns the evaluation criteria and the agent acts as proxy. The relevant question is, *Does the agent's behavior deliver value to the user per the user's standard?*

When a strongly autonomous agent is used in an application it must simultaneously fill both roles. In effect, it is caught in a pun on the word "agent". It is an agent in the sense of tool, and agent in the sense of actor or principle. In this situation the relevant evaluation question is, *Will the agent deliver value to a user as a consequence of delivering value to itself?*

The answer to this question will be complicated by the fact that the user and agent perceive and act on the world in different terms, possess different background knowledge, and will at best own imperfectly matching value functions. Thus, the consideration of strong autonomy highlights frame comparison problems.

We submit that frame comparison issues exist in all agent evaluation contexts, but that they are rarely, if ever, explicitly addressed.

Key Questions

- ***What computational architecture supports strong autonomy?***
  - agent must nominate, select, and abandon tasks in service of own value
  - agent must learn from real world reward
- ***What principles constrain agent-held values?***
- ***How do we use a strongly autonomous agent?***
  - encode a base of skills
  - program via values instead of goals
  - provide run-time advice vs commands
- ***What design methodology ensures an agent will deliver value to users by delivering value to itself?***
  - need a mathematics that spans reference frames

6. Key Questions

Our sketch for the form and function of strong autonomy raises several key technical questions. The most obvious is, *How do we build it?* The problem is non-trivial since a strongly autonomous agent requires novel abilities: at minimum, it must be able to nominate objectives for consideration in a situation relevant way, then select and arbitrate amongst them when resource contention occurs. In addition, the agent should only pursue objectives that yield highest reward, and abandon those that prove poor generators. Our approach relies heavily on learning technology to synchronize agent expectations with results and to map situations into suggested objectives. We employ and extend decision theoretic structures to represent plans and quantify value.

The strong autonomy framework also appears to change the programming and user interaction metaphors. Our intuition is that an engineer should provide the agent with a vocabulary of skills, and an application designer should shape the agent's behavior by instilling a particular value function. The user will then communicate with the agent to influence it at run time. Note that a strongly autonomous agent can *only* act in service of its interests. This implies that users can inform, instruct, and forcefully suggest options, but they can only *command* this type of agent at some cost to the metaphor.

The question concerning design methodology is critical to the strong autonomy model; at stake is our ability to design, employ, and control such agents to perform desired functions. We seek a formal mathematical model that supports behavioral guarantees. As mentioned in the previous section, this will require mathematics for comparing reference frames together with a method of projecting future state. Our work to date employs a probabilistic mapping between agent and user state, and Markov models for projection.

In order to gift a strongly autonomous agent with fluid set of core values we need a principle that governs value change. This question is far more philosophical than technical since it asks, *What should an agent hold dear?* Related questions are, *How can an agent come to value something new?*, and *Should some priorities (e.g. survival) remain fixed?* Our intuition is to pursue a constructivist and functional tack; the agent should value what it can obtain and devalue what it cannot. It should be well-adjusted. One can accept or reject the principle but the consequences are worth exploring. Depending upon the environment, the principle could produce (in metaphor) a hopeless drudge or a manic achiever. Either is perfectly functional. The design question is, *Which is better?*

Work to Date

- **Implemented an artificial agent with its own value structure (based on Icarus system [Langley])**
  - competence: plan (build decision tree), decide (max EV), measure, learn (update EV)
  - tested in four domains (pole balancing, truck backing, flight control, real time flight control)
  - *demonstrated agent improved own value by giving up on user-supplied tasks*
  - *extended decision theory to incorporate experience*

- **Defined an iterative agent design methodology**
  - relates agent and user perceptions of state
  - ascends gradient of user utility wrt agent mods

[Schoppers, M., and Shapiro, D., *Designing Embedded Agents to Optimize End-User Objectives*, Intelligent Agents IV, Springer Verlag, to appear]

7. Work to Date

As of this writing we have substantively addressed two of the four critical issues named in the previous section. We have implemented a prototype of a strongly autonomous agent which possesses its own values and selects its own objectives, and we have constructed a design methodology based on the comparison of agent and user reference frames. We have not yet addressed the user interaction metaphor or implementation of the "well-adjustedness" principle.

Our agent prototype puts a number of key features in place; the system can build plans for objectives, calculate expected values, choose courses of action by a value maximization principle, measure received reward, and adapt its expected values as received reward diverges. We have tested this agent in four domains and demonstrated an unusual ability; the agent increases its received value by giving up on user-supplied tasks. En route, we extended Decision Theory by introducing a notion of experience distinct from expectation, and by defining the appropriate inter-relations. This enabled us to compare expected values with received rewards on strong theoretic grounds. (The Appendix contains a paper which documents this system.)

Our work on design methodology employs a simplified model; it treats the agent purely as a tool without an internal value structure, but it *does* address cross-frame comparison. In specific, we generate a conditional probability table relating agent and user perceptions of state, and we employ it to transform projected agent states into projected user corollaries. This leads to an iterative improvement methodology which ascends the gradient of user utility by employing targeted agent design changes. (We report on this work in the Appendix. It was not funded by this research grant but it is relevant to the investigation of strong autonomy.)

Conclusions to Date

- ***Goals come from Values***
  - goals are elements of plans
  - plans are errorfull models for obtaining reward
  - agents nominate & select plans by expected value; they abandon (or persist in) plans in response to received reward.

- ***Values come from interaction with the world***
  - agents learn expected values from received reward
  - *but what received value function is appropriate?*

- ***Agents should sense value-laden world features***
  - above and beyond those necessary for action

8. Conclusions to Date

Several conclusions emerge from our work to date. The first is that goals plausibly come from values. The agent we implemented employs values to drive planning in a computationally efficient way, and again to select maximizing action. Since real world plans are neither sound nor complete (due to the lack of a closed world assumption), agents require two species of values; one associated with anticipation and the other with experience. This leads to our second conclusion: values plausibly come from interaction with the world. In particular while anticipated value drives action selection, received value (through adaptation) adjusts anticipation, and therefor drives task abandonment and persistence. We will take this point further as work on adjusting the received value function progresses.

We offer one more conclusion concerning selective sensing which we also studied in the context of this project (see the Appendix). Agents should sense value-laden features of the world, not just those necessary for action. This step supports agent self-evaluation, specifically the ability to detect flaws in plans (viewed as generators of value vs. future state). This conclusion generalizes [Brook's] very influential suggestion to severely restrict sensing by coupling it with action routines.

Future Work

- ***Reimplement architecture***
- ***Demonstrate programming by value***
  - test ease of specifying aggressive/passive drivers
  - evaluate model fidelity
- ***Define and test a well-adaptedness principle***
  - value what you achieve, devalue what you cannot
  - seek $\dfrac{\text{expected value}}{\text{received reward}} \cong 1$
- ***Expand design methodology***
  - add agent held values, non-Markovian user utility
  - determine value of communicating a distinction
  - examine computational efficiency
- ***Build and validate a strongly autonomous agent***

9. Future Work

Our future work plan examines those features of strong autonomy we have yet to address. The initial step, however, is to reimplement the current architecture in light of our experience. Among other changes, we are making the agent more reactive by incorporating logical, optional and sequential structure to its plans. Our formalism is based on Universal Plans [Schoppers] and Teleo-Reactive [Nilsson] with slight generalizations. The second implementation will also clarify the decision theoretic structure by allowing values to be attached to outcomes as well as processes.

Work on programming by value addresses changes to the user interaction metaphor induced by strong autonomy (see Section 6). We are shifting to an automotive domain with less real-time pressure than aircraft control, and plan to realize aggressive and passive drives by making localized changes to the agents received value function. The agents will reuse the same skill base. We hope to evaluate the resulting models against a user-held utility function. That is, we will determine if the agent shares the same relative preference over situation-action pairs it and the user simultaneously perceive.

Our work on a well-adaptedness principle is in progress. While our governing intuitions are reasonably clear we have just now established candidate mechanisms. Our current approach (1) learns to value unvalued distinctions by spreading received value across co-occurring features, and (2) it learns core preferences by anchoring on average received value and increasing (or decreasing) feature weights whenever received reward is greater (or less than) average. Average results produce no change, while a sustained hostile environment will decrease rewards, and with some delay the underlying anchor. As this reduced reward feeds back into expectations, the agent will prefer qualitatively different actions.

We have identified several plausible extensions to our work on design methodology. The first is to model the agent as a Markov process *with* reward, and calculate the value the agent will deliver to the user while pursuing value for itself. We can then improve a given value driven design by hill-climbing in user utility over changes to the agent's actions (as before). A second idea is to calculate the value of communicating a distinction to the agent as measured by the expected increase in user utility. By extension, if all user-held distinctions were available, the agent should be capable of user-quality behavior. Less endowed agents can be compared against this gold standard. We also plan to characterize the computational efficiency of the calculations which support the design methodology. Our current results suggest we can treat agents with 10^6 states, but this claim should be examined in more general contexts.

The main objective of our near term work is to demonstrate a full-pass through the design and validation process. Our goal is to build a strongly autonomous agent, estimate the value it will provide to a user, and show how to improve such a design in an incremental and principled way. This accomplishment would open the door to the application of strongly autonomous systems.

References

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation, 2*, 14-23

Kushmeric, N., Hanks, S., & Weld, D., An algorithm for probabilistic planning. *AI Journal 76:1*, 1995, 239--286.

Nilsson, N.J. (1994). Teleoreactive programs for agent control. *Journal of Artificial Intelligence Research, 1*, 139-158.

Schoppers, M., (1995). The use of dynamics in an intelligent controller for a space-faring rescue robot. *Artificial Intelligence Journal ,73,* 175-230

Appendix

We have included the following five papers as part of this report:

Langley, P., *Learning to sense selectively in physical domains*, Proceedings of the First International Conference on Autonomous Agents (1997), Marina del Rey, CA.

Langley, P., *An abstract computational model of learning selective sensing skills*, Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society (1996), San Diego, CA.

Shapiro, D., *Giving up for no good reason,* 19th Annual Conference of the Cognitive Science Society (1997), Stanford, CA (poster).

Shapiro, D., *Giving up by losing interest,* unpublished.

Schoppers, M., & Shapiro, D., *Designing embedded agents to optimize end-user-objectives*, Proceedings Agent Theories Architecture and Languages Workshop (1997), Providence, RI also to appear in Intelligent Agents IV, Springer Verlag.

The first four were funded in whole or in part by this contract. The last is included for its relevance to the topic of strong autonomy.