Toward Trustworthy Adjustable Autonomy in KAoS

Jeffrey M. Bradshaw, Hyuckchul Jung, Shri Kulkarni, Matthew Johnson,
Paul Feltovich, James Allen, Larry Bunch, Nathanael Chambers, Lucian Galescu,
Renia Jeffers, Niranjan Suri, William Taysom, & Andrzej Uszok

Institute for Human and Machine Cognition (IHMC), 40 S. Alcaniz
Pensacola, FL 32502 - USA
{jbradshaw, hjung, skulkarni, mjohnson, pfeltovich, jallen, lbunch,
nchambers, lgalescu, rjeffers, nsuri, wtaysom, auszok}@ihmc.us
http://www.ihmc.us

Abstract. Trust is arguably the most crucial aspect of agent acceptability. At its simplest level, it can be characterized in terms of judgments that people make concerning three factors: an agent's competence, its benevolence, and the degree to which it can be rapidly and reliably brought into compliance when things go wrong. Adjustable autonomy consists of the ability to dynamically impose and modify constraints that affect the range of actions that the human-agent team can successfully perform, consistently allowing the highest degrees of useful autonomy while maintaining an acceptable level of trust. Many aspects of adjustable autonomy can be addressed through policy. Policies are a means to dynamically regulate the behavior of system components without changing code or requiring the cooperation of the components being governed. By changing policies, a system can be adjusted to accommodate variations in externally imposed constraints and environmental conditions. In this paper we describe some important dimensions relating to autonomy and give examples of how these dimensions might be adjusted in order to enhance performance of human-agent teams. We introduce Kaa (KAoS adjustable autonomy) and provide a brief comparison with two other implementations of adjustable autonomy concepts.

1 Introduction

As computational systems with increasing autonomy interact with humans in more complex ways—and with the welfare of the humans sometimes dependent on the conduct of the agents—there is a natural concern that the agents act in predictable ways so that they will be acceptable to people [6].

Trust is arguably the most crucial aspect of agent acceptability. As Alan Kay has written: “It will not be an agent’s manipulative skills, or even its learning abilities, that will get it accepted, but instead its safety and ability to explain itself in critical situations.… At the most basic level the thing we want most to know about an agent is not how powerful it can be, but how trustable it is” [36, pp. 205-206].[1]

[1] As an interesting sideline reflecting the importance of trust in cooperative relationships among people, a USA Today poll published in February 2003 noted that about two-thirds of Ameri-

The concept of trust, as it applies to agent systems, is a very complex topic whose theory and practice has been extensively studied (e.g., [21]). At its most basic level, trust can be characterized in terms of judgments that people make concerning three factors: an agent's competence, its benevolence, and the degree to which it can be rapidly brought into compliance when things go wrong.[2]

Competence is the ability to reliably perform some task in a manner that is consistent with expectations and requirements. In artificial systems, competence is typically assured through a variety of engineering practices. Unfortunately, many important capabilities (including some of the most basic human capabilities such as vision) are currently beyond our power to effectively engineer and implement. And even for those kinds of systems we know how to build, our foresight is limited with respect to unexpected circumstances that may render an otherwise competent system impotent in a particular application context.

A judgment of system *benevolence* is based upon our confidence that it is free from malicious intent. Developers commonly attempt to assure this quality by restricting reliance on system components and information to those coming from trusted sources, and rejecting elements of unknown or untrusted provenance. However, current trends in development practices complicate our ability to assure protection from malicious intent in this fashion, as it becomes increasingly rare to engineer complex systems completely in house from scratch. Moreover, the open nature of the Internet increasingly requires interaction with unknown people and computing entities of all kinds.

When all else fails, we depend on measures assuring a system's *compliance* with supervisory control to make up for gaps in its competence and to limit damage from malicious intent. Such control is typically attempted through various forms of human monitoring and intervention. However human resources, human ability, and human attention span may be too limited to make this a practical solution. Moreover, highly-complex systems are often designed and coupled in ways that make them prone to subtle cascading failures ("normal accidents" [46]) that unyieldingly and sometimes disastrously resist human attempts to wrest back control of system operations in critical situations.

In everyday life, trust of both people and of engineered systems is built through the synergistic processes of observation and explanation. With time and experience the observer learns to distinguish between the situations where the subject of observation is likely to act competently and benevolently, and those where it is not. Pertinent and accurate explanation of the subject's actions can speed up this process of learning by observation. Through experience in seeing the results of providing outside direction to the subject in order to avoid or to recover from failure (whether such failure is inadvertent or intentional) the observer also has an opportunity to learn something about the subject's disposition for compliance: proving the technology to see whether it will do all things that it is commanded.

Because their motives and behavior are autonomous and complex, trust building among people in matters of non-trivial concern can take a long time. At the other

cans pick "trustworthiness" as the most important factor—more important than community knowledge, work history, or personality—in their choice of a real estate agent.

[2] Competence and benevolence as primary dimensions in human attribution of personal qualities are discussed in [14].

extreme, people will more readily confide tasks to simple deterministic mechanisms whose design is artfully made transparent.[3]

Agents and other autonomous systems occupy a strange middle ground between such extremes, which sometimes makes their acceptance by people difficult [6; 44]. On the one hand, their autonomy and intelligence grants agents the flexibility and additional competence needed to handle challenging situations that require significant "wiggle room" for self-governed actions. On the other hand (given the significance of the tasks with which they are commonly entrusted) an agent's potential for blindness to the limits of its competence, for non-transparent complexity, and for inadequate controllability can be a formula for disaster. We need a way to "bound the wiggle room" of the agents so that their degrees of autonomy are consistent with human judgments about their trustworthiness.

Policies are a means to dynamically regulate the behavior of a system without changing code or requiring the cooperation of the components being governed. They can be used to address the three aspects of trust mentioned:

- Through policy, people can precisely express bounds on autonomous behavior in a way that is consistent with their appraisal of an agent's *competence* in a given context.
- Because policy enforcement is handled externally to the agent, malicious and buggy agents can no more exempt themselves from the constraints of policy than *benevolent* and well-written ones can.
- The ability to change policies dynamically means that poorly performing agents can be immediately brought into *compliance* with corrective measures.

Elsewhere we have pointed out other benefits of policy-based approaches, including reusability, efficiency, extensibility, context-sensitivity, verifiability, support for both simple and sophisticated components, and reasoning about component behavior [6].

In the mid-1990's, we began to define the initial version of KAoS, a set of platform-independent services that enable people to define policies ensuring adequate predictability and controllability of both agents and traditional distributed systems [11; 13; 40; 52; 55]. Since that time, we have also become involved in a series of projects requiring close and continuous interaction among humans and agents in military and space settings. In collaboration with our research partners, we have been developing a generic model of human-agent teamwork that includes policies to assure natural and effective interaction in mixed teams of people and robots [1; 5; 9; 48]. As part of this effort, we have argued that policies have important analogues in animal societies and human cultures that can be exploited in the design of artificial systems [24]. So far, so good.

What was still lacking was a means to enable policies to be adjusted without requiring a human in the loop as circumstances change. Such a capability for "adjustable autonomy" would amount to an automated way to "wiggle the bounds of the

[3] Effective user interfaces often take advantage of the ontological expectations that users bring with them when they interact with various portrayals of functionality in graphical user interfaces. The illusion of simplicity thus created can be helpful in building user trust and understanding so long as these expectations are not violated.

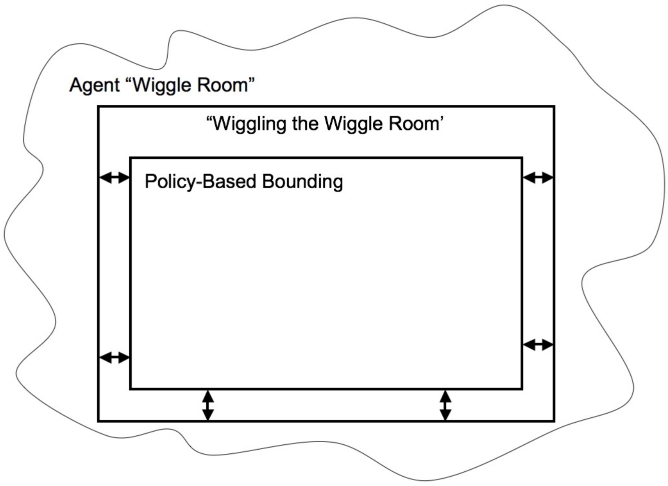


Fig. 1. An agent's "wiggle" room consists of its set of performable actions in a given context while the policy-based bounds that people impose on that wiggle room define a smaller region of trusted operation. Capabilities for adjustable autonomy support the modification of these bounds at runtime in order to adapt to changing conditions.

wiggle room" (see figure 1). To be sure, this hoped-for gain in adaptivity would mean some loss in predictability. Moreover, second-order issues of limited competence would no doubt now emerge at the level of the component doing the adjusting. Despite these challenges we believe that a well-tuned adjustable autonomy component can be of great value for many applications.

According to our view, adjustable autonomy consists of the ability to *dynamically impose and modify constraints that affect the range of actions that the human-agent team can successfully perform, consistently allowing the highest degrees of useful autonomy while maintaining an acceptable level of trust*. Though adjustable autonomy is hardly a new topic in agent systems,[4] there has been a general lack of consensus on terminology and basic concepts. Moreover, current approaches have been based on simplistic assumptions about the nature of human-automation interaction that are generally not informed by the lessons learned from decades of research in human factors and the behavioral and social sciences.

In subsequent sections, we describe the multi-dimensional nature of adjustable autonomy as we construe it (section 2) and give examples of how various dimensions might be adjusted in order to enhance performance of human-agent teams (section 3). We then introduce Kaa, the KAoS adjustable autonomy component (section 4). Finally, we provide a brief comparison of alternate extant approaches to adjustable autonomy (section 5), and offer some concluding remarks (section 6).

[4] Similarly-motivated research has also been undertaken with respect to more general automation issues under the label of "dynamic function allocation" [29].

2 Some Dimensions of Autonomy

The word "autonomy," which is straightforwardly derived from a combination of Greek terms signifying self-government (*auto-* (self) + *nomos* (law)) has two basic senses in everyday usage.[5]. In the first sense, we use the term to denote *self-sufficiency,* the capability of an entity to take care of itself. This sense is present in the French term *autonome* when, for example, it is applied to someone who is successfully living away from home for the first time. The second sense refers to the quality of *self-directedness,* or freedom from outside control, as we might say of a portion of a country that has been identified as an "autonomous region."[6]

Some important dimensions relating to autonomy can be straightforwardly characterized by reference to figure 2.[7] Note that the figure does not show every possible configuration of the dimensions, but rather exemplifies a particular set of relations holding for the actions of a particular set of actors in a given situation. There are two basic dimensions:

- a *descriptive* dimension corresponding to the first sense of autonomy (self-sufficiency) that stretches horizontally to describe the actions an actor in a given context is *capable* of performing; and
- a *prescriptive* dimension corresponding to the second sense of autonomy (self-directedness) running vertically to describe the actions an actor in a given context is allowed to perform or which it must perform by virtue of *policy* constraints in force.

The outermost rectangle, labeled *potential actions,* represents the set of all actions across all situations defined in the ontologies currently in play.[8] Note that there is no

[5] Here we are only concerned with those dimensions that seem directly relevant to adjustable autonomy as we define it. Some excellent detailed and comprehensive analyses of the concept of autonomy that go beyond what can be treated in this paper have been collected in [30; 39].

[6] We note that "no man [or machine] is an island"—and in this sense of reliance and relation to others, complete autonomy is a myth.

[7] See [9] for a more complete discussion of these dimensions, and their relationship to mixed-initiative interaction. Much of sections 2 and 3 are adapted from this chapter. We can make a rough comparison between some of these dimensions and the aspects of autonomy described by Falcone and Castelfranchi [23]. Environmental autonomy can be expressed in terms of the possible actions available to the agent—the more the behavior is wholly deterministic in the presence of a fixed set of environmental inputs, the smaller the range of possible actions available to the agent. The aspect of self-sufficiency in social autonomy relates to the ranges of what can be achieved independently vs. in concert with others; deontic autonomy corresponds to the range of permissions and obligations that govern the agent's choice among actions.

[8] The term *ontology* is borrowed from the philosophical literature, where it describes a theory of what exists. Such an account would typically include terms and definitions only for the very basic and necessary categories of existence. However, the common usage of ontology in the knowledge representation community is as a vocabulary of representational terms and their definitions at any level of generality. A computational system's "ontology" defines what exists for the program—in other words, what can be represented by it. It should be observed that we speak deliberately in terms of actions and not in terms of goals or objectives—

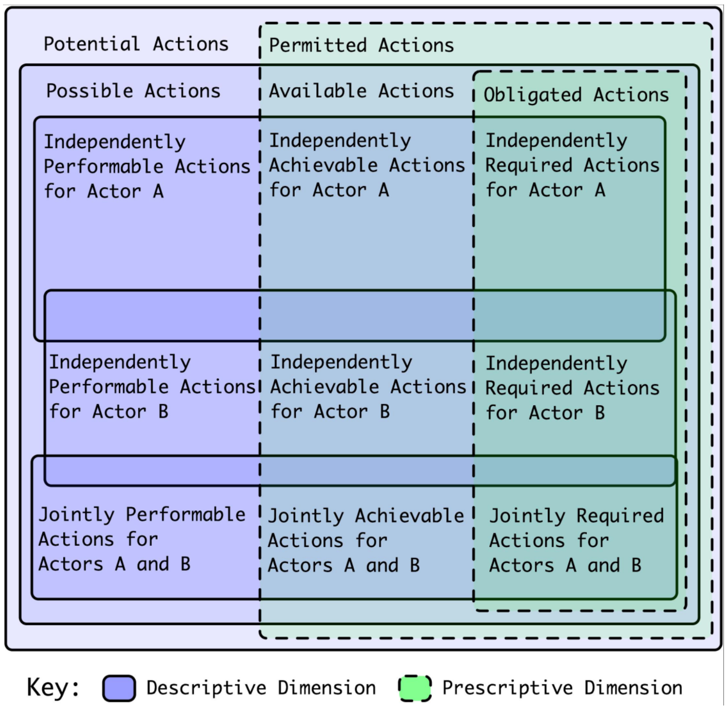

Potential Actions
Permitted Actions
Possible Actions
Available Actions
Obligated Actions
Independently Performable Actions for Actor A
Independently Achievable Actions for Actor A
Independently Required Actions for Actor A
Independently Performable Actions for Actor B
Independently Achievable Actions for Actor B
Independently Required Actions for Actor B
Jointly Performable Actions for Actors A and B
Jointly Achievable Actions for Actors A and B
Jointly Required Actions for Actors A and B
Key:
Descriptive Dimension
Prescriptive Dimension

Fig. 2. Some dimensions of autonomy.

requirement that every action in the unknowable and potentially chaotic *universe of actions* that a set of actors may take be represented in the ontology; only those which are of consequence for adjustable autonomy need be included.

The rectangle labeled *possible actions* represents the set of potential actions whose performance by one or more actors is deemed plausible in a given situation [3; 20].[9] Note that the definition of possibilities is strongly related to the concept of affordances [27; 43], in that it relates the features of the situation to classes of actors capable of exploiting these features in the performance of actions.[10]

Of these possible actions, only certain ones will be deemed *performable* for a given actor[11] (e.g., Actor A) in a given situation. *Capability,* i.e., the power that makes an action performable, is a function of the *abilities* (e.g., knowledge, capacities, skills) and *conditions* (e.g., ready-to-hand resources) necessary for an actor to successfully undertake some action in a given context. Certain actions may be *independently per-*

we do not assume that the system adjusting autonomy has access to goals and objectives, only that it can regulate observable actions in the world.

[9] The evaluation of possibility, necessarily a subjective judgment, admits varying degrees of confidence—for example, one can distinguish mere plausibility of an action from a more studied feasibility. These nuances of possibility are not discussed in this paper.

[10] As expressed by Norman: “Affordances reflect the possible relationships among actors and objects: they are properties of the world” [45].

[11] For purposes of discussion, we use the term *actor* to refer to either a biological entity (e.g., human, animal) or an artificial agent (e.g., software agent, robotic agent).

formable by either Actor A or B; other actions can be independently performed by either one or the other uniquely.[12] Yet other actions are *jointly performable* by a set of actors.

Along the prescriptive dimension, declarative policies may specify various *permissions* and *obligations* [19]. An actor is *free* to the extent that its actions are not limited by permissions or obligations. *Authorities* may impose or remove involuntary policy constraints on the actions of actors.[13] Alternatively, actors may voluntarily enter into *agreements* that mutually bind them to some set of policies for the duration of the agreement. The *effectivity* of an individual policy specifies when it is in or out of force.

The set of *permitted actions* is determined by *authorization policies* that specify which actions an actor or set of actors is allowed (*positive authorizations* or *A+* policies) or not allowed (*negative authorizations* or *A-* policies) to perform in a given context.[14] The intersection of what is possible and what is permitted delimits the set of *available actions*.

Of those actions that are available to a given actor or set of actors, some subset may be judged to be *independently achievable* in the current context. Some actions, on the other hand, would be judged to be only *jointly achievable*.

Finally, the set of *obligated actions* is determined by *obligation policies* that specify actions that an actor or set of actors is required to perform (*positive obligations* or *O+* policies) or for which such a requirement is waived (*negative obligations* or *O-* policies).[15] *Jointly obligated actions* are those that two or more actors are explicitly required to perform.

3 Adjustable Autonomy

A major challenge in the design of intelligent systems is to ensure that the degree of autonomy is continuously and transparently adjusted in order to meet the performance expectations imposed by the system designer and the humans and agents with which the system interacts. We note that is not the case that "more" autonomy is always

[12] Although we show A and B sharing the same set of possible actions, this need not always be the case. Also, note that in our example the range of jointly achievable actions has overlap only with Actor B and not Actor A.

[13] Authority relationships may be, at the one extreme, static and fixed in advance and, at the other, determined by negotiation and persuasion as the course of action unfolds.

[14] We note that some permissions (e.g., network bandwidth reservations) involve allocation of finite and/or consumable resources, whereas others do not (e.g., access control permissions). We also note that obligations (see below) typically require allocation of finite abilities and resources; when obligations are no longer in effect, these abilities and resources may become free for other purposes.

[15] A negative obligation corresponds to the idea of "you are not obliged to" rather than "you are obliged not to"—this second sense corresponds to a negative authorization with the subject doing the enforcing (similar to Ponder's *refrain* policies [19]).

better:[16] as with a child left unsupervised in city streets during rush hour, an unsophisticated actor insufficiently monitored and recklessly endowed with unbounded freedom may pose a danger both to others and to itself. On the other hand, a capable actor shackled with too many constraints will never realize its full potential.

Thus, a primary purpose of adjustable autonomy is to maintain the system being governed at a sweet spot between convenience (i.e., being able to delegate every bit of an actor's work to the system) and comfort (i.e., the desire to not delegate to the system what it can't be trusted to perform adequately).[17] The coupling of autonomy with policy mechanisms gives the agent maximum freedom for local adaptation to unforeseen problems and opportunities while assuring humans that agent behavior will be kept within desired bounds. If successful, adjustable autonomy mechanisms give the added bonus of assuring that the definition of these bounds can be appropriately responsive to unexpected circumstances.

All this, of course, only complicates the agent designer's task, a fact that has lent urgency and impetus to efforts to develop broad theories and general-purpose frameworks for adjustable autonomy that can be reused across as many agents, domains, and applications as possible. To the degree that adjustable autonomy services can be competently implemented and packaged for convenient use within popular development platforms, agent designers can focus their attention more completely on the unique capabilities of the individual agents they are developing, while relying on the extant services to assist with addressing cross-cutting concerns about human-agent interaction.

We now consider some of the dimensions on which autonomy can be adjusted.

Adjusting Permissions. A first case to consider is that of adjusting permissions. Reducing permissions may be useful when it is concluded, for example, that an agent is habitually attempting actions that it is not capable of successfully performing—as when a robot continues to rely on a sensor that has been determined to be faulty. It may also be desirable to reduce permissions when agent deliberation about (or execution of) certain actions might incur unacceptable costs or delays.

If, on the other hand, an agent is known to be capable of successfully performing actions that go beyond what it is currently permitted to do, its permissions could be increased accordingly. For example, a flying robot whose duties had previously been confined to patrolling the space station corridors for atmospheric anomalies could be given additional permissions allowing it to employ its previously idle active barcode sensing facilities to take equipment inventories while it is roaming [12; 26].

Adjusting Obligations. On the one hand, "underobligated" agents can have their obligations increased—up to the limit of what is achievable—through additional task assignments. For example, in performing joint action with people, they may be obliged to report their status frequently or to receive explicit permission from a human before proceeding to take some action. On the other hand, an agent should not be

[16] In fact, the multidimensional nature of autonomy argues against even the effort of mapping the concept of "more" and "less" to a single continuum. See [22] for an overview of a broad theory of adjustable autonomy and its multi-dimensional nature.

[17] We note that reluctance to delegate can also be due to other reasons. For example, some kinds of work may be enjoyable to people—such as skilled drivers who may prefer a manual to an automatic transmission.

required to perform any action that outstrips its permissions, capabilities, or possibilities.[18] An "overcommitted" agent can sometimes have its autonomy adjusted to manageable levels through reducing its current set of obligations. This can be done through delegation, facilitation, or renegotiation of obligation deadlines. In some circumstances, the agent may need to renege on its obligations in order to accomplish higher priority tasks.

Adjusting Possibilities. A highly capable agent may sometimes be performing below its capabilities because of restrictions on resources available in its current situation. For example, a physical limitation on network bandwidth available through the nearest wireless access point may restrict an agent from communicating at the rate it is permitted and capable of doing.[19]

In some circumstances, it may be possible to adjust autonomy by increasing the set of possibilities available to an agent. For example, a mobile agent may be able to make what were previously impossible faster communication rates possible by moving to a new host in a different location. Alternatively, a human could replace an inferior access point with a faster one.

Sometimes reducing the set of possible actions provides a powerful means of enforcing restrictions on an agent's actions. For example, an agent that "misbehaved" on the network could be sanctioned and constrained from some possibilities for action by moving it to a host with restricted network access.

Adjusting Capabilities. The capabilities of an agent affect the range of its performable actions. In this sense, the autonomy of an agent can be augmented either by increasing its own independent capabilities or by extending its joint capabilities through access to other actors to which tasks may be delegated or shared. An agent's capabilities can also be affected indirectly by adjusting possibilities in a way that changes current conditions (e.g., externally adding or reducing needed resources) or directly by, for example, reallocating one's internal resources and efforts.

An adjustable autonomy service aimed at increasing an agent's capabilities could assist in discovering agents with which an action that could not be independently achieved could be jointly achieved. Or if the agent was hitting the ceiling on some computational resource (e.g., bandwidth, memory), resource access policies could be adjusted to allow the agent to leverage the additional assets required to perform some action. Finally, the service could assist the agent by facilitating the deferral, delegation, renegotiation, or reneging on obligations in order to free up previously committed resources (as previously mentioned in the context of adjusting obligations).

[18] In some cases, rather than rejecting commitments to unachievable obligations outright, it may be preferable to increase permissions, capabilities, or possibilities (if possible), thus transforming an unachievable obligation into one that is achievable. It is also thinkable that someone may wish to obligate an agent to do something beyond its individual capabilities—this might be called enforced cooperation.

[19] Besides constrained resources, other features of the situation may also limit the possibility of certain actions, e.g., the darkness of nighttime may prevent me from reading.

Having described the principal dimensions of autonomy and the kinds of adjustments that can be made, we now apply that perspective to the implementation of these ideas.[20]

4 Kaa: KAoS Adjustable Autonomy

We are currently working to develop and evaluate formalisms and mechanisms for adjustable autonomy and policies that will facilitate effective coordination and mixed-initiative interaction among humans and agents engaged in joint activities. We are doing this in conjunction with a testbed that integrates the various capabilities of TRIPS, Brahms, and KAoS [5].

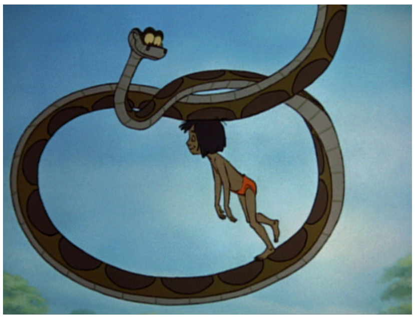


Fig. 3. In Kipling's *Jungle Book,* Kaa rescued Mowgli from harm.

KAoS is a collection of componentized policy and domain services.[21] KAoS policy services enable the specification, management, conflict resolution, and enforcement of semantically-rich policies defined in OWL [54].[22] On this foundation, we are building Kaa (KAoS adjustable autonomy) a component that permits KAoS to perform automatic adjustments of autonomy consistent with policy.[23]

[20] In this paper, we do not discuss the close relationship between adjustable autonomy and mixed-initiative interaction. A discussion of our views can be found in [9], and the broader theoretical context of coordination of joint human-agent activity is given in [10; 37; 38].

[21] KAoS is compatible with several popular agent frameworks, including Nomads [49], the DARPA CoABS Grid [35], the DARPA ALP/UltraLog Cougaar framework (http://www.cougaar.net) [40], CORBA (http://www.omg.org), Voyager (http://www.recursionsw.com/osi.asp), Brahms (www.agentisolutions.com) [48], TRIPS [2; 5], and SFX (http://crasar.eng.usf.edu/research/publications.htm). While initially oriented to the dynamic and complex requirements of software agent applications, KAoS services are also being adapted to general-purpose grid computing (http://www.gridforum.org) and Web Services (http://www.w3.org/2002/ws/) environments as well [34; 55]. KAoS has been deployed in a wide variety of applications, from coalition warfare [12; 50] and agile sensor feeds [50], to process monitoring and notification [15], to robustness and survivability for distributed systems [40], to semantic web services composition [57], to human-agent teamwork in space applications [12], to cognitive prostheses for augmented cognition [6].

[22] Going beyond OWL-DL, we have made a few judicious extensions to description logic within KAoS (e.g., role-value maps) [57].

[23] At first glance, it may seem paradoxical that Kaa would be both subject to policy and be able to adjust policy (and other autonomy dimensions). By representing Kaa as a subject of KAoS policy, we can establish the bounds that govern the operations of Kaa to make sure that it does not make the kinds of adjustments that people do not want it or trust it to make on its own.

In Rudyard Kipling's Jungle Book, the human boy Mowgli was educated in the ways and secrets of the jungle by Kaa the python. His hypnotic words and stare charmed the malicious monkey tribe that had captured the boy, and Kaa's encircling coils at last “bounded" their actions and put an end to their misbehavior.[24] In a similar way, Kaa attempts to bound the autonomy of agents (see figure 3).

Assistance from Kaa in making autonomy adjustments might typically be required when it is anticipated that the current configuration of human-agent team members has led to or is likely to lead to failure, and when there is no set of competent and authorized humans available to make the adjustments themselves. Ultimately, the value of performing an adjustment in a given context is a matter of expected utility: the utility of making the change vs. the utility of maintaining the status quo.

The current implementation of Kaa uses influence-diagram-based decision-theoretic algorithms to determine what if any changes should be made in agent autonomy [7; 8; 33]. However, Kaa is designed to allow other kinds of decision-making components to be plugged-in if an alternative approach is preferable. When invoked, Kaa first compares the utility of various adjustment options (e.g., increases or decreases in permissions and obligations, acquisition of capabilities, proactive changes to the situation to allow new possibilities), and then—if a change in the status quo is warranted—takes action to implement the recommended alternative.

When evaluating options for adaptively reallocating tasks among team members, Kaa should consider that dynamic role adjustment comes at a cost. Hence, measures of expected utility would ideally be used in the future to evaluate the tradeoffs involved in potentially interrupting the ongoing activities of agents and humans in such situations to communicate, coordinate, and reallocate responsibilities [18; 31; 32].

Ultimately, it would also be important for Kaa to consider that the need for adjustments may cascade in complex fashion: interaction may be spread across many potentially distributed agents and humans who act in multiply connected interaction loops. For this reason, adjustable autonomy may involve not merely a shift in roles among a human-agent pair, but rather the distribution of dynamic demands across many coordinated actors.

Finally, as Hancock and Scallen [29] rightfully observe, the problem of adaptive function allocation is not merely one of efficiency or technical elegance. Economic factors (e.g., can the task be more inexpensively performed by humans, agents, or some combination?), political and cultural factors (e.g., is it acceptable for agents to perform tasks traditionally assigned to humans?), or personal and moral considerations (e.g., is a given task enjoyable and challenging vs. boring and mind-numbing for the human?) are also essential considerations.

To the extent circumstances allow Kaa to adjust agent autonomy with reasonable dynamism (ideally allowing handoffs of control among team members to occur anytime) and with a sufficiently fine-grained range of levels, teamwork mechanisms can flexibly renegotiate roles and tasks among humans and agents as needed when new opportunities arise or when breakdowns occur. Such adjustments can also be anticipatory when agents are capable of predicting the relevant events [4; 23].

[24] A somewhat different Kaa character and story was later portrayed in the Disney movie.

4.1. A Simple Example: Robot Signaling

One of the most important contributions of more than a decade of research on agent teamwork is the finding that many aspects of effective team behavior rely on a collection of generic coordination mechanisms rather than on deep knowledge of specific application domains [17; 53]. With previous research in agent teamwork, we share the assumption that, to the extent possible, teamwork knowledge should be modeled explicitly and separately from the problem-solving domain knowledge so it can be easily reused across applications. In such an approach, policies for agent safety and security (as well as contextual and culturally sensitive teamwork behavior) can be represented as KAoS policies that enable many aspects of the nature and timing of the agent's interaction with people to be appropriate, without requiring each agent to individually encode that knowledge [6].

As part of this research, we are developing policies to govern various nonverbal forms of expression in software agents and robots [24]. Such nonverbal behaviors are intended to express not only the current state of the agent but also—importantly—to provide rough clues about what it is going to do next. In this way, people can be better enabled to participate with the agent in coordination, support, avoidance, and so forth. In this sense, nonverbal expressions are an important ingredient in enabling human-agent teamwork. A simple example involving a nonverbal expression policy will illustrate a simplified description of how Kaa works.

Assume that a robot's signaling behavior is governed by the following positive obligation policy: *O+: A robot must beep for a few seconds before beginning to move*. The intention of such a policy is to warn others nearby to stay out of the way when a robot is about to move (see figure 4).

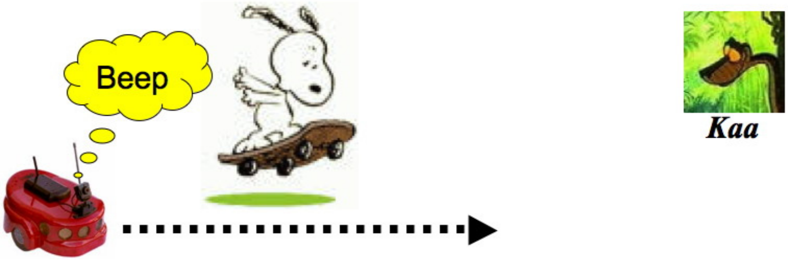
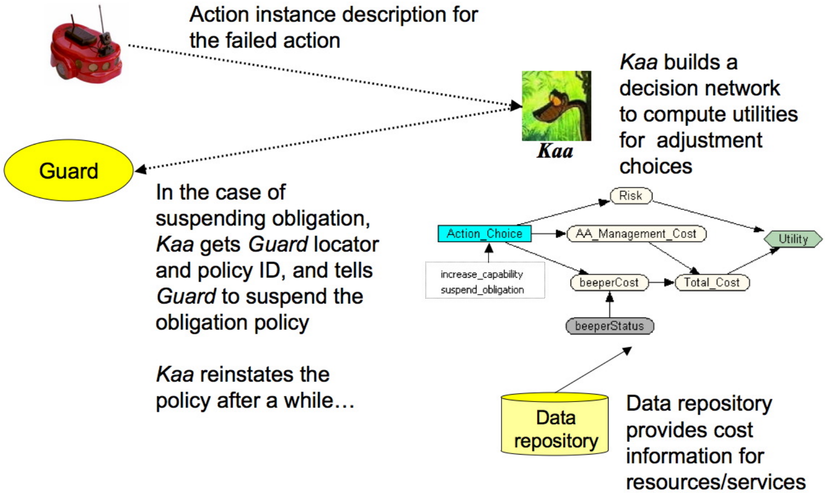


Fig. 4. Kaa stands ready to intervene in case of failure of the warning beeper,

Before the robot attempts to move, the robot execution platform, in conjunction with platform-specific KAoS components, requires the robot to ask a KAoS guard responsible for managing local policy enforcement whether the action is authorized.[25] The guard then retrieves and checks the relevant set of policies. In this example, we assume that the guard finds both an authorization policy allowing the robot to move in this context as well as the obligation policy described above. Under normal circumstances, the obligation policy will first trigger the robot to emit the beep, and then will return the necessary authorization for the robot to move. However, certain states and

[25] KAoS policy enforcement is described in more detail in [56].


Action instance description for the failed action

Kaa builds a decision network to compute utilities for adjustment choices

In the case of suspending obligation, *Kaa* gets *Guard* locator and policy ID, and tells *Guard* to suspend the obligation policy

Kaa reinstates the policy after a while…

Data repository provides cost information for resources/services

Fig. 5. Kaa concept of operation for the robot beep failure example,

events, such as a failure of the robot to successfully sound its obligatory warning, will trigger an attempt by Kaa to intervene in a helpful way.[26]

In such a case, the KAoS-Robot infrastructure creates an action instance description for the failed action and forwards it to Kaa (figure 5). Kaa in turn dynamically constructs an influence diagram based on state-specific information in the action instance description combined, when network availability allows, with information from the KAoS directory service repository.

After considering available alternatives (e.g., increasing the range of performable actions vs. decreasing the range of obliged actions), let's assume that Kaa determines that temporarily suspending the obligation policy is the best option. With this precondition for the move action now removed, the guard can now return its authorization for the move to the robot, and the robot can perform the action. When circumstances permit, Kaa can reinstate the suspended policy.

4.2. Application to Office of Naval Research-Sponsored Research

The ONR-sponsored Naval Automation and Information Management Technology (NAIMT) project is a collaborative effort of the Naval Surface Warfare Center, Panama City (NSWC PC), IHMC, and the University of South Florida (USF) to integrate key technologies to meet the military's future needs for coordinating the operation of unmanned systems with greater effectiveness and affordability. Unmanned systems will play an increasing role in military actions. Large numbers of unmanned ground, air, underwater, and surface vehicles will work together, coordinated by ever smaller teams of human operators. In order to be operationally efficient, effective and useful, these robots must perform complex tasks with considerable autonomy, must work

[26] Alternately, Kaa could be configured to watch for component failures in advance and take preemptive actions before the failure occurs.

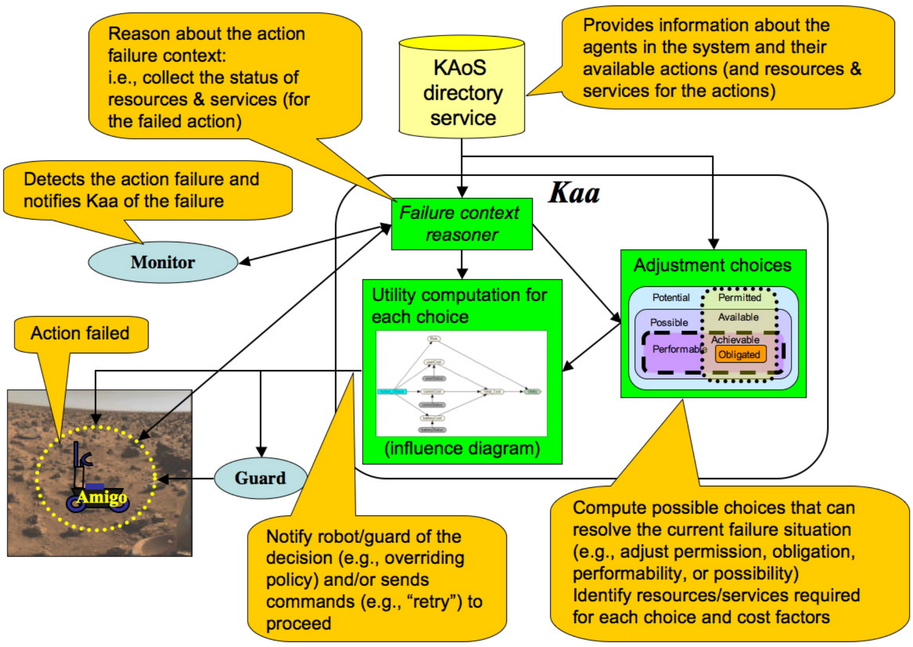

Reason about the action failure context:
i.e., collect the status of resources & services (for the failed action)
KAoS directory service
Provides information about the agents in the system and their available actions (and resources & services for the actions)
Detects the action failure and notifies Kaa of the failure
Kaa
Failure context reasoner
Monitor
Adjustment choices
Potential
Permitted
Possible
Available
Achievable
Performable
Obligated
Utility computation for each choice
(influence diagram)
Action failed
Amigo
Guard
Notify robot/guard of the decision (e.g., overriding policy) and/or sends commands (e.g., "retry") to proceed
Compute possible choices that can resolve the current failure situation (e.g., adjust permission, obligation, performability, or possibility)
Identify resources/services required for each choice and cost factors

Fig. 6. Notional functional architecture for Kaa,

together safely and reliably within policy constraints, must operate flexibly and robustly in the face of intermittent network availability and potentially rapid fluctuation of available infrastructure resources, and must coordinate their actions with each other and with human operators. In addition, the human operator, controlling the actions of many unmanned systems must observe and control them in an intuitive fashion incorporating capabilities for mixed-initiative interaction and adjustable autonomy.

A current demonstration scenario is based on a lane clearing operation in shallow water. Using cooperative search algorithms developed jointly by NSWC PC and IHMC, lane free of mines are identified in order to allow the landing of amphibious vehicles on the beach. IHMC's Agile Computing Infrastructure provides the communication and computation framework, including ad-hoc networking, reliable communication over mobile ad-hoc networks, opportunistic resource discovery and exploitation, and flexible, bandwidth-efficient data feeds [50; 51]. IHMC's TRIPS component addresses the challenges of providing an effective and natural multimodal interface (including spoken dialog) between the human operator(s) and the robotic platforms [2; 16; 25]. KAoS provides the policy management services that govern the behavior of the robotic platforms, the data flows within the agile computing infrastructure, and adjustment of autonomy among the robotic platforms and human operators. KAoS also provides policies operating in conjunction with the USF SFX architecture (http://crasar.eng.usf.edu/research/publications.htm).

Figure 6 shows a notional functional architecture for Kaa, while figure 7 shows its relationships to other components of the demonstration. Unlike the simplified example presented in the previous section, either Kaa or a human operator or both can potentially intervene to assist the human-robot team when necessary. Preferences for

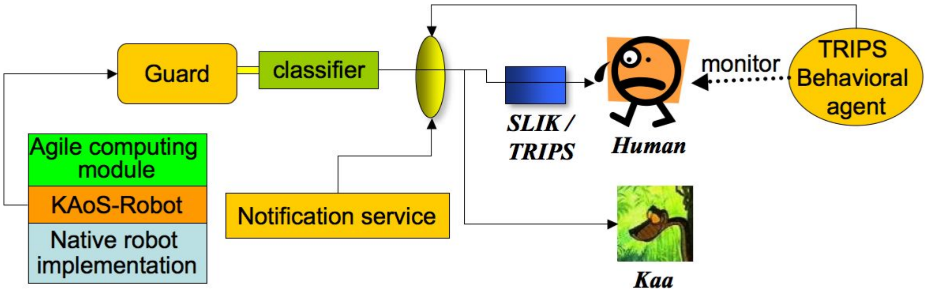


Fig. 7. Relationships of Kaa to other NAIMT application components,

who should intervene can be expressed on a policy-by-policy basis. Thus, in some situations the person defining the policy may feel comfortable always letting Kaa handle problems on its own without interrupting the operator. In other situations, the person may only trust the human to intervene. In yet other situations, the person defining the policy may want to give the operator the first opportunity to intervene and only if the operator is too busy to respond will it call on Kaa for help. Finally, a policy may be specified that requires Kaa to make the first attempt at resolving any problems, allowing it, however, to call upon a human for help if it deems necessary.[27]

For example, one constant challenge our project has been asked to address is autonomous restoration of lost network connectivity among the robots. In such cases, the agile computing infrastructure frequently tasks an idle robot to move into a position where it can serve as a network relay. Sometimes for one reason or another, a robot is not authorized by policy to make a given move. Rather than simply turning down the authorization, the guard will forward the request to a classifier. The classifier examines the policy to determine who (if anyone) should be consulted in such circumstances. Normally, the classifier will forward the request to the human, who will decide if the need for restoring network connectivity should override the policy restriction. The notification service, in conjunction with the TRIPS behavioral agent, determines the means by which the human should be contacted and the urgency with which it should be presented given an understanding of the state of the human. If after a timeout period the human does not reply, the decision about whether to grant permission can be delegated to Kaa.

In another example, a policy requires the robot to contact the human for help when the robot is not certain about the identification of a mine. With multiple robots moving and many tasks to monitor, it is possible that a robot will find an indeterminate object while the operator is occupied with other tasks. If the operator fails to respond within sufficient time to such a query, a request for help is forwarded to Kaa. After evaluating alternatives Kaa may determine to grant the robot the autonomy it needs to mark the indeterminate object as a mine on its own and move on. Alternatively, Kaa may determine to extend the timeout period, up to some maximum allowed it by policy.

[27] SLIK (Simple Logical Interface to KAoS) provides a custom interface between TRIPS and KAoS. KAoS-Robot provides an ontology-based layer of abstraction for various robot implementations and assists the guard in policy enforcement.

4.3. Future Work

Our initial experiences with Kaa as part of the NAIMT project have explored only a fraction of the issues involved in applying a richer model of adjustable autonomy to human-agent interaction. Two ambitious areas for future work are briefly discussed below.

General decision models. Our aim in developing Kaa is not to create a series of point solutions for individual applications, but rather to create and validate a general model for adjustable autonomy that can work in tandem with KAoS teamwork policies across a wide spectrum of domains. Moving beyond the application-specific influence diagrams we have constructed for NAIMT demonstrations, we intend to represent the specific implications of our general model of adjustable autonomy in skeletal knowledge bases *(influence diagram templates)* of probabilistic information, alternatives, and preferences that can be reused for particular classes of decisions. Portions of this knowledge base would then be combined with application-specific and situation-specific information, alternatives, and preferences obtained at runtime.

Performance metrics and utility functions. For specific applications, we need to better understand the relative contribution and interrelationships among different performance metrics that could be used to evaluate the overall results of using Kaa.[28] A similar understanding is needed in order to specify appropriate utility functions on which Kaa can base a given decision. Some of the many dimensions that could be considered include: *survivability* (ability to maintain effectiveness in the face of unforeseen software or hardware failures), *safety* (ability to prevent certain classes of dangerous actions or situations), *predictability* (assessed correlation between human judgment of predicted vs. actual behavior), *controllability*(effectiveness and immediacy with which an authorized human can prevent, stop, enable, or initiate agent actions), *effectiveness* (assessed correlation between human judgment of desired vs. actual behavior), and *adaptability* (ability to respond to changes in context).

5 A Comparison of Perspectives

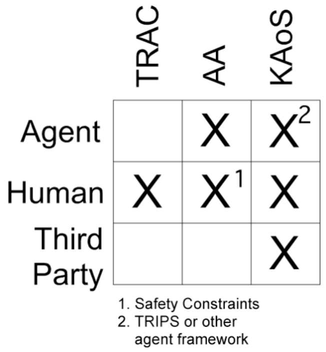
Several groups have grappled with the problem of characterizing and developing practical approaches for implementing adjustable autonomy in deployed systems. Each takes a little different approach and uses similar terminology somewhat differently. It would be helpful to the research community if there were an increased consensus about the concepts and terminology involved.

To characterize a sampling of perspectives and terminology used by various research groups, we will briefly contrast our approach to two other implemented formulations: the SRI TRAC[29] framework [42] and the Electric Elves agent-based autonomy framework [47]. These two frameworks were compared in [41], making them a convenient choice for further comparison.

Those wishing a comprehensive review of these two frameworks should consult the above-referenced publications. Here we will ignore specific features of the

[28] See [18; 28; 31] for a sampling of perspectives on this issue.

[29] Now reimplemented under the name of SPARK.



	TRAC	AA	KAoS
Agent		X	X[2]
Human	X	X[1]	X
Third Party			X

1. Safety Constraints
2. TRIPS or other agent framework

Fig. 8. Party taking the initiative for adjustment

frameworks (e.g., analysis tools, user interface, accommodation of multiple agents from heterogeneous platforms) as well as performance and scalability issues, and will only consider some of the basic dimensions relating to the adjustment decision:

- Party taking initiative for adjustment
- Rationale for considering adjustment
- Type of adjustment
- Default modality
- Duration of adjustment
- Party who is final arbiter
- Locus of enforcement.

Party taking initiative for adjustment. In principle, the actual adjustment of an agent's level of autonomy could be initiated either by a human, the agent, or some other software component. Figure 8 illustrates how this is handled in the three frameworks.[30]

TRAC has been characterized as a framework for "user-based adjustable autonomy" in which policies are defined by people. The motivation for these policies is to compensate for limits to agent competence and to allow for personalization.

[30] Cohen [18] draws a line between those approaches in which the agent itself wholly determines the mode of interaction with humans (mixed-initiative) and those where this determination is imposed externally (adjustable autonomy). Additionally, mixed-initiative systems are considered by Cohen to generally consist of a single user and a single agent. However, it is clear that we take the position that these two approaches are not mutually exclusive and that, in an ideal world, agents would be capable of both reasoning about when and how to initiate interaction with the human and also of subjecting themselves to the external direction of whatever set of explicit authorization and obligation policies were currently in force to govern that interaction.

Additionally, there is no reason to limit the notion of "mixed initiative" systems to the single agent-single human case. Hence we prefer to think of mixed-initiative systems as being those systems that are capable of making context-appropriate adjustments to their level of social autonomy (i.e., their level or mode of engagement with the human), whether a given adjustment is made as a result of reasoning internal to the agent or due to externally-imposed policy-based constraints.

The Electric Elves approach, on the other hand, has been characterized as an "agent-based autonomy" (AA) approach where adjustments to autonomy are the result of explicit agent reasoning. A *transfer-of-control* strategy is computed in advance and offline. It is implemented using a Markov decision process (MDP) such that in each possible state the agent knows whether it should make the decision autonomously, ask the user for help, or change its coordination constraints (e.g., inform other agents of a delay). Humans can also define "safety constraints" (see below).

Since KAoS runs in conjunction with several agent frameworks, the ability for an agent to explicitly reason about autonomy adjustment depends on the particular platform being used. For example, TRIPS allows sophisticated reasoning about these issues, whereas agents built with less capable frameworks could be very simple. In KAoS, humans can define or change policies through a simple graphical user interface called KPAT. Additionally, Kaa, as a selectively trusted third-party, can sometimes make its own adjustments to policy or other dimensions of autonomy.

Rationale for considering adjustment. Many different factors can constitute the rationale for considering an adjustment.

In TRAC, the rationale for modification to policy resides exclusively with the human, whereas in AA it is part of a precomputed set of agent strategies, with choices determined according to a fixed set of agent states.

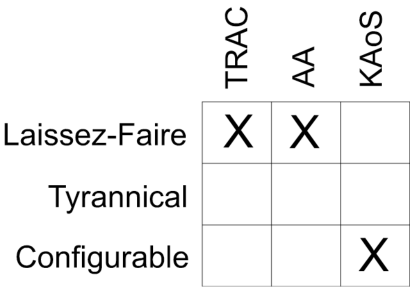
In KAoS, authorized people or agents can make changes to policy at any time. In addition, any event or state in the world or the ontology that can be monitored by the system could be set up to trigger a self-adjustment process in Kaa. For example, the impetus for Kaa to consider adjustment could be due to the fact that task performance has fallen outside of (or has returned within) some acceptable range. Alternatively, certain events or changes in the state of the environment (e.g., sudden change in temperature), an agent (e.g., agent is performing erratically), or a human (e.g., human is injured; or conversely is now again available to help out) can provide the rationale for adjustment.

Type of adjustment. As outlined in section 2, adjustments to autonomy can be of several types: capabilities (more or less), possibilities, authorizations (positive or negative, more or less), and obligations (positive or negative, more or less).[31]

TRAC allows policies to be defined for three sorts of positive obligations: obligations to ask permission from a human supervisor for certain actions *(permission requirements),* obligations to defer decisions about certain actions to a human supervisor *(consultation requirements),* and obligations to accomplish specified tasks in a certain manner *(strategy preference guidance).*

AA allows for agents to determine strategy for and require itself to act upon three kinds of obligations: asking the user for help, making the decision itself, and performing a coordinating action. Additionally, AA allows the human to represent two kinds of safety constraints. The first kind is a sort of negative authorization that can prevent agents from taking a given action, while the second kind is a sort of positive obligation that can require them to take a given action.

[31] Though it is fair to characterize TRAC as an implementation of an approach to adjustable autonomy, it should be pointed out that the ability to allow humans to define policies is different from the *automatically* adjustable autonomy implemented by AA and Kaa.



	TRAC	AA	KAoS
Laissez-Faire	X	X	
Tyrannical			
Configurable			X

Fig. 9. Default modality,

KAoS is designed to allow adjustment along any of the dimensions described in section 2.

Default modality. Within a given policy-governed environment, a default modality for authorization policies must be established. In a permissive environment, it is usually easiest to set a permissive default modality and to define a small number of negative authorization policies for any actions that are restricted. In a restrictive environment, the opposite is usually true.

TRAC and AA implement a fixed *laissez-faire* modality where anything is permitted that is not specifically forbidden by policy (figure 9).

KAoS implements a per-domain-configurable default modality. In other words, for a given application, actors in one domain (i.e., user-defined group) might be subject to a *laissez-faire* default modality, while actors in another domain might be simultaneously subject to a *tyrannical* one (i.e., where everything is forbidden that is not specifically permitted). Modality dominance constraints are used to determine which modality takes priority in the case of actors belonging to more than one domain.

Duration of adjustment. When constraints in any of the three frameworks are put into force or removed, the adjustment to the agent's level of autonomy is changed indefinitely. However, KAoS additionally allows an authorized human or trusted software component such as Kaa to override current policy on a per event basis (e.g., exceptionally allow some action just this once) or for a certain fixed length of time (e.g., allow some action for the next one hour).

Party who is final arbiter. For different classes of action, there is the ultimate question of who is the final authority in case of disagreement about authorizations between some person and the agent. For example, a UAV may have the policy that a human can always take manual control if there is a risk of an imminent crash. On the other hand, the UAV may have a policy that prevents a human from ever deliberately crashing the UAV.

In TRAC, this issue does not arise, because it is not possible to represent authorization policies. In AA and KAoS, authorization policies can limit the kinds of actions that the agent can perform. Additionally, in KAoS, authorization policies can limit human actions as well.

Locus of enforcement. In both TRAC and AA, the interpretation of policies is integrated with the agent's planning and decision-making process and the agent itself is entrusted with the enforcement of policy.

While KAoS does not prohibit agents from optimizing their behavior through reasoning about policies (to the extent that policy disclosure is itself permitted by policy), the responsibility for enforcement is given to independent control elements of the trusted infrastructure. In this way, enforcement of policy remains effective even when agents themselves are buggy, malicious, poorly designed, or unsophisticated. This is essential if policies are to be regarded as something binding on agents, rather than just good advice. This being said, there is no reason why KAoS enforcement mechanisms could not be used in complementary fashion with the agent-based enforcement approach in TRAC and AA.

6 Concluding Observations

We believe that policy-based approaches hold great promise in compensating for limitations of competence, benevolence, and compliance of agent systems. Semantically-rich policy representations like those used in KAoS enable flexibility, extensibility, and power for policy specification, modification, reasoning, and enforcement.

As the work in this chapter demonstrates, the application of policy is now being extended beyond narrow technical concerns, such as security, to social aspects of trust and human-agent teamwork. As research results bring greater experience and understanding of how to implement self-regulatory mechanisms for agent systems, we expect a convergence and a concomitant increase of synergy among researchers with differing perspectives on adjustable autonomy and mixed-initiative interaction.

One of the biggest areas of difference between KAoS and the two other approaches compared in section 5 (TRAC and AA) is in where the locus of initiative for adjustment and enforcement lies. Though allowing policy to be disclosed and reasoned about by agents when required, KAoS policy services aim to assure that policy can be relied on whether or not the agents themselves can be trusted to do the right thing. In contrast, TRAC and AA depend exclusively on the agents to monitor and enforce their own actions. Thus when humans are presumed to be more trustworthy than the agents themselves, the KAoS policy enforcement approach would seem to have merit. However, manual policy specification using KAoS is insufficient for those situations where the human is unavailable or is judged to be less competent or trustworthy than the machine for dealing with an adjustable autonomy issue. Our objective in developing Kaa is to address this issue: to enable reasoning about relevant tradeoffs and the taking of appropriate measures in situations where the best action may not be the blind following of a policy but rather the adjustment of one or more dimensions of autonomy. While we have reservations about approaches that *cannot* enforce human-defined policies independently of a potentially untrustworthy or incompetent agent's code, we also have qualms about approaches lacking the means to adjust policies and policy-related autonomy dimensions that have been clearly demonstrated to be *ineffective* in a given context of application. Adding the capabilities of Kaa to KAoS services is intended to achieve the best of both worlds: trustworthy adjustable autonomy regardless of the trustworthiness of agent code.

Acknowledgements

The authors gratefully acknowledge the support of the DARPA EPCA (CALO), DAML, and Ultra*Log Programs, the NASA Cross-Enterprise and Intelligent Systems Programs, the Army Research Lab, and the Office of Naval Research while preparing this paper. We are also grateful for the contributions of Guy Boy, Kathleen Bradshaw, Maggie Breedy, Mark Burstein, Alberto Cañas, Bill Clancey, Ken Ford, Mike Goodrich, Mark Greaves, David Gunning, Jack Hansen, Pat Hayes, Robert Hoffman, Gary Klein, Henry Lieberman, James Lott, Rebecca Montanari, Luc Moreau, Robin Murphy, Don Norman, Dan Olsen, Jerry Pratt, Ken Seamons, Mike Shafto, Maarten Sierhuis, Niranjan Suri, Milind Tambe, Austin Tate, Gianluca Tonti, Gary Toth, Larry Tokarcik, Ron Van Hoof, Rob Winkler, David Woods, and Tim Wright.

References

[1] Acquisti, A., Sierhuis, M., Clancey, W. J., & Bradshaw, J. M. (2002). Agent-based modeling of collaboration and work practices onboard the International Space Station. *Proceedings of the Eleventh Conference on Computer-Generated Forces and Behavior Representation*. Orlando, FL,

[2] Allen, J. F., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L., & Stent, A. (2001). Towards conversational human-computer interaction. *AI Magazine*, 22(4), 27-35.

[3] Barwise, J., & Perry, J. (1983). *Situations and Attitudes*. Cambridge, MA: MIT Press.

[4] Boella, G. (2002). Obligations and cooperation: Two sides of social rationality. In H. Hexmoor, C. Castelfranchi, & R. Falcone (Ed.), *Agent Autonomy*. (pp. 57-78). Dordrecht, The Netherlands: Kluwer.

[5] Bradshaw, J. M., Acquisti, A., Allen, J., Breedy, M. R., Bunch, L., Chambers, N., Feltovich, P., Galescu, L., Goodrich, M. A., Jeffers, R., Johnson, M., Jung, H., Lott, J., Olsen Jr., D. R., Sierhuis, M., Suri, N., Taysom, W., Tonti, G., & Uszok, A. (2004). Teamwork-centered autonomy for extended human-agent interaction in space applications. *AAAI 2004 Spring Symposium*. Stanford University, CA, AAAI Press,

[6] Bradshaw, J. M., Beautement, P., Breedy, M. R., Bunch, L., Drakunov, S. V., Feltovich, P. J., Hoffman, R. R., Jeffers, R., Johnson, M., Kulkarni, S., Lott, J., Raj, A., Suri, N., & Uszok, A. (2004). Making agents acceptable to people. In N. Zhong & J. Liu (Ed.), *Intelligent Technologies for Information Analysis: Advances in Agents, Data Mining, and Statistical Learning*. (pp. 361-400). Berlin: Springer Verlag.

[7] Bradshaw, J. M., & Boose, J. H. (1990). Decision analysis techniques for knowledge acquisition: Combining information and preferences using *Aquinas* and *Axotl*. *International Journal of Man-Machine Studies*, 32(2), 121-186.

[8] Bradshaw, J. M., Covington, S. P., Russo, P. J., & Boose, J. H. (1990). Knowledge acquisition for intelligent decision systems: integrating *Aquinas* and *Axotl* in DDUCKS. In M. Henrion, R. Shachter, L. N. Kanal, & J. Lemmer (Ed.), *Uncertainty in Artificial Intelligence*. (pp. 255-270). Amsterdam: Elsevier.

[9] Bradshaw, J. M., Feltovich, P., Jung, H., Kulkarni, S., Taysom, W., & Uszok, A. (2004). Dimensions of adjustable autonomy and mixed-initiative interaction. In M. Nickles, M. Rovatsos, & G. Weiss (Ed.), *Agents and Computational Autonomy: Potential, Risks, and Solutions. Lecture Notes in Computer Science, Vol. 2969*. (pp. 17-39). Berlin, Germany: Springer-Verlag.

[10] Bradshaw, J. M., Feltovich, P. J., Jung, H., Kulkarni, S., Allen, J., Bunch, L., Chambers, N., Galescu, L., Jeffers, R., Johnson, M., Sierhuis, M., Taysom, W., Uszok, A., & Van Hoof, R. (2004). Policy-based coordination in joint human-agent activity. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. The Hague, Netherlands,

[11] Bradshaw, J. M., Greaves, M., Holmback, H., Jansen, W., Karygiannis, T., Silverman, B., Suri, N., & Wong, A. (1999). Agents for the masses: Is it possible to make development of sophisticated agents simple enough to be practical? *IEEE Intelligent Systems*(March-April), 53-63.

[12] Bradshaw, J. M., Sierhuis, M., Acquisti, A., Feltovich, P., Hoffman, R., Jeffers, R., Prescott, D., Suri, N., Uszok, A., & Van Hoof, R. (2003). Adjustable autonomy and human-agent teamwork in practice: An interim report on space applications. In H. Hexmoor, R. Falcone, & C. Castelfranchi (Ed.), *Agent Autonomy*. (pp. 243-280). Kluwer.

[13] Bradshaw, J. M., Suri, N., Breedy, M. R., Canas, A., Davis, R., Ford, K. M., Hoffman, R., Jeffers, R., Kulkarni, S., Lott, J., Reichherzer, T., & Uszok, A. (2002). Terraforming cyberspace. In D. C. Marinescu & C. Lee (Ed.), *Process Coordination and Ubiquitous Computing*. (pp. 165-185). Boca Raton, FL: CRC Press. Updated and expanded version of an article that originally appeared in IEEE Intelligent Systems, July 2001, pp. 49-56.

[14] Brown, B. L., & Bradshaw, J. M. (1985). A psychology of vocal patterns. In H. Giles & R. N. S. Clair (Ed.), *Language and the Paradigms of Social Psychology*. Hillsdale, N.J.: Lawernce Erlbaum.

[15] Bunch, L., Breedy, M. R., & Bradshaw, J. M. (2004). Software agents for process monitoring and notification. *Proceedings of AIMS 04*.

[16] Chambers, N., Allen, J., & Galescu, L. (2005). A dialogue-based approach to multi-robot team control. *Proceedings of Third International Naval Research Labs Multi-Robot Systems Workshop*. Washington, D.C.,

[17] Cohen, P. R., & Levesque, H. J. (1991). *Teamwork*. Technote 504. Menlo Park, CA: SRI International, March.

[18] Cohen, R., & Fleming, M. (2002). Adjusting the autonomy in mixed-initiative systems by reasoning about interaction. In H. Hexmoor, C. Castelfranchi, & R. Falcone (Ed.), *Agent Autonomy*. (pp. 105-122). Dordrecht, The Netherlands: Kluwer.

[19] Damianou, N., Dulay, N., Lupu, E. C., & Sloman, M. S. (2000). *Ponder: A Language for Specifying Security and Management Policies for Distributed Systems, Version 2.3*. Imperial College of Science, Technology and Medicine, Department of Computing, 20 October 2000.

[20] Devlin, K. (1991). *Logic and Information*. Cambridge, England: Cambridge University Press.

[21] Falcone, R., Barber, S., Korba, L., & Singh, M. (Ed.). (2003). *Trust, Reputation, and Security: Theories and Practice (LNAI 2631)*. Berlin: Springer.

[22] Falcone, R., & Castelfranchi, C. (2002). Adjustable social autonomy.

[23] Falcone, R., & Castelfranchi, C. (2002). From automaticity to autonomy: The frontier of artificial agents. In H. Hexmoor, C. Castelfranchi, & R. Falcone (Ed.), *Agent Autonomy*. (pp. 79-103). Dordrecht, The Netherlands: Kluwer.

[24] Feltovich, P., Bradshaw, J. M., Jeffers, R., Suri, N., & Uszok, A. (2004). Social order and adaptability in animal and human cultures as an analogue for agent communities: Toward a policy-based approach. In *Engineering Societies in the Agents World IV. LNAI 3071*. (pp. 21-48). Berlin, Germany: Springer-Verlag.

[25] Ferguson, G., & Allen, J. (1998). TRIPS: An integrated intelligent problem-solving assistant. *Proceedings of the National Conference on Artificial Intelligence (AAAI 98)*. Madison, WI,

[26] Gawdiak, Y., Bradshaw, J. M., Williams, B., & Thomas, H. (2000). R2D2 in a softball: The Personal Satellite Assistant. H. Lieberman (Ed.), *Proceedings of the ACM Conference*

on Intelligent User Interfaces (IUI 2000), (pp. 125-128). New Orleans, LA, New York: ACM Press,

[27] Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Boston, MA: Houghton Mifflin.

[28] Guinn, C. I. (1999). Evaluating mixed-initiative dialog. *IEEE Intelligent Systems*, September-October, 21-23.

[29] Hancock, P. A., & Scallen, S. F. (1998). Allocating functions in human-machine systems. In R. Hoffman, M. F. Sherrick, & J. S. Warm (Ed.), *Viewing Psychology as a Whole*. (pp. 509-540). Washington, D.C.: American Psychological Association.

[30] Hexmoor, H., Falcone, R., & Castelfranchi, C. (Ed.). (2003). *Agent Autonomy*. Dordrecht, The Netherlands: Kluwer.

[31] Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. Pittsburgh, PA, New York: ACM Press,

[32] Horvitz, E., Jacobs, A., & Hovel, D. (1999). Attention-sensitive alerting. *Proceedings of the Conference on Uncertainty and Artificial Intelligence (UAI '99),* (pp. 305-313). Stockholm, Sweden,

[33] Howard, R. A., & Matheson, J. E. (1984). Influence diagrams. In R. A. Howard & J. E. Matheson (Ed.), *Readings on the Principles and Applications of Decision Analysis*. (pp. 719-762). Menlo Park, California: Strategic Decisions Group.

[34] Johnson, M., Chang, P., Jeffers, R., Bradshaw, J. M., Soo, V.-W., Breedy, M. R., Bunch, L., Kulkarni, S., Lott, J., Suri, N., & Uszok, A. (2003). KAoS semantic policy and domain services: An application of DAML to Web services-based grid architectures. *Proceedings of the AAMAS 03 Workshop on Web Services and Agent-Based Engineering*. Melbourne, Australia,

[35] Kahn, M., & Cicalese, C. (2001). CoABS Grid Scalability Experiments. O. F. Rana (Ed.), *Second International Workshop on Infrastructure for Scalable Multi-Agent Systems at the Fifth International Conference on Autonomous Agents*. Montreal, CA, New York: ACM Press,

[36] Kay, A. (1990). User interface: A personal view. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design*. (pp. 191-208). Reading, MA: Addison-Wesley.

[37] Klein, G., Feltovich, P. J., Bradshaw, J. M., & Woods, D. D. (2004). Common ground and coordination in joint activity. In W. B. Rouse & K. R. Boff (Ed.), *Organizational Simulation*. (pp. in press). New York City, NY: John Wiley.

[38] Klein, G., Woods, D. D., Bradshaw, J. M., Hoffman, R., & Feltovich, P. (2004). Ten challenges for making automation a "team player" in joint human-agent activity. *IEEE Intelligent Systems*, 19(6), 91-95.

[39] Klusch, M., Weiss, G., & Rovatsos, M. (Ed.). (2004). *Computational Autonomy*. Berlin, Germany: Springer-Verlag.

[40] Lott, J., Bradshaw, J. M., Uszok, A., & Jeffers, R. (2004). Using KAoS policy and domain services within Cougaar. *Proceedings of the Open Cougaar Conference 2004,* (pp. 89-95). New York City, NY,

[41] Maheswaran, R. T., Tambe, M., Varakantham, P., & Myers, K. (2004). Adjustable autonomy challenges in personal assistant agents: A position paper. In M. Klusch, G. Weiss, & M. Rovatsos (Ed.), *Computational Autonomy*. (pp. in press). Berlin, Germany: Springer.

[42] Myers, K., & Morley, D. (2003). Directing agents. In H. Hexmoor, C. Castelfranchi, & R. Falcone (Ed.), *Agent Autonomy*. (pp. 143-162). Dordrecht, The Netherlands: Kluwer.

[43] Norman, D. A. (1988). *The Psychology of Everyday Things*. New York: Basic Books.

[44] Norman, D. A. (1997). How might people interact with agents? In J. M. Bradshaw (Ed.), *Software Agents*. (pp. 49-55). Cambridge, MA: The AAAI Press/The MIT Press.

[45] Norman, D. A. (1999). Affordance, conventions, and design. *Interactions*, May, 38-43.

[46] Perrow, C. (1984). *Normal Accidents: Living with High-Risk Technologies.* New York: Basic Books.

[47] Scerri, P., Pynadath, D., & Tambe, M. (2002). Adjustable autonomy for the real world. In H. Hexmoor, C. Castelfranchi, & R. Falcone (Ed.), *Agent Autonomy.* (pp. 163-190). Dordrecht, The Netherlands: Kluwer.

[48] Sierhuis, M., Bradshaw, J. M., Acquisti, A., Van Hoof, R., Jeffers, R., & Uszok, A. (2003). Human-agent teamwork and adjustable autonomy in practice. *Proceedings of the Seventh International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS).* Nara, Japan,

[49] Suri, N., Bradshaw, J. M., Breedy, M. R., Groth, P. T., Hill, G. A., Jeffers, R., Mitrovich, T. R., Pouliot, B. R., & Smith, D. S. (2000). NOMADS: Toward an environment for strong and safe agent mobility. *Proceedings of Autonomous Agents 2000.* Barcelona, Spain, New York: ACM Press,

[50] Suri, N., Bradshaw, J. M., Carvalho, M., Breedy, M. R., Cowin, T. B., Saavendra, R., & Kulkarni, S. (2003). Applying agile computing to support efficient and policy-controlled sensor information feeds in the Army Future Combat Systems environment. *Proceedings of the Annual U.S. Army Collaborative Technology Alliance (CTA) Symposium.*

[51] Suri, N., Carvalho, M., & Bradshaw, J. M. (2004). Proactive resource management for agile computing. C. Bryce & G. Czaijkowski (Ed.), *Proceedings of the Tenth Annual ECOOP Workshop on Mobile Object Systems and Resource-Aware Computing.* Oslo, Norway,

[52] Suri, N., Carvalho, M., Bradshaw, J. M., Breedy, M. R., Cowin, T. B., Groth, P. T., Saavendra, R., & Uszok, A. (2003). Mobile code for policy enforcement. *Policy 2003.* Como, Italy,

[53] Tambe, M., Shen, W., Mataric, M., Pynadath, D. V., Goldberg, D., Modi, P. J., Qiu, Z., & Salemi, B. (1999). Teamwork in cyberspace: Using TEAMCORE to make agents team-ready. *Proceedings of the AAAI Spring Symposium on Agents in Cyberspace.* Menlo Park, CA, Menlo Park, CA: The AAAI Press,

[54] Tonti, G., Bradshaw, J. M., Jeffers, R., Montanari, R., Suri, N., & Uszok, A. (2003). Semantic Web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder. In D. Fensel, K. Sycara, & J. Mylopoulos (Ed.), *The Semantic Web—ISWC 2003. Proceedings of the Second International Semantic Web Conference, Sanibel Island, Florida, USA, October 2003, LNCS 2870.* (pp. 419-437). Berlin: Springer.

[55] Uszok, A., Bradshaw, J. M., Jeffers, R., Johnson, M., Tate, A., Dalton, J., & Aitken, S. (2004). Policy and contract management for semantic web services. *AAAI 2004 Spring Symposium Workshop on Knowledge Representation and Ontology for Autonomous Systems.* Stanford University, CA, AAAI Press,

[56] Uszok, A., Bradshaw, J. M., Jeffers, R., Suri, N., Hayes, P., Breedy, M. R., Bunch, L., Johnson, M., Kulkarni, S., & Lott, J. (2003). KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. *Proceedings of Policy 2003.* Como, Italy,

[57] Uszok, A., Bradshaw, J. M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., & Aitken, S. (2004). KAoS policy management for semantic web services. *IEEE Intelligent Systems*, 19(4), 32-41.