# Dimensions of Adjustable Autonomy and Mixed-Initiative Interaction

Jeffrey M. Bradshaw, Paul J. Feltovich, Hyuckchul Jung, Shriniwas Kulkarni, William Taysom, and Andrzej Uszok

Institute for Human and Machine Cognition (IHMC),
40 S. Alcaniz, Pensacola, FL 32502
{jbradshaw, pjfeltovich, hjung, skulkarni, wtaysom, auszok}@ihmc.us
http://www.ihmc.us

**Abstract**. Several research groups have grappled with the problem of characterizing and developing practical approaches for implementing adjustable autonomy and mixed-initiative interaction in deployed systems. However, each group takes a little different approach and uses variations of the same terminology in a somewhat different fashion. In this chapter, we will describe some common dimensions in order to better understand these important but ill-characterized topics. We will also sketch the approach to implementation we are developing in the context of our research on policy-governed autonomous systems.

## 1 Introduction

As computational systems with increasing autonomy interact with humans in more complex ways, there is a natural concern that they are sufficiently predictable and controllable as to be acceptable to people [5]. In addition to traditional concerns for safety and robustness in such systems, there are important social aspects relating to mutual situation awareness, intent recognition, coordination of joint tasks, and efficiency and naturalness of the interaction that must be attended to [6; 15]. Since autonomous entities cannot always be trusted to regulate their own behavior appropriately, various approaches have been proposed to allow continuous external adjustment of the bounds of autonomous behavior, assuring their ongoing safety and effectiveness.

Policies are a powerful means for dynamically regulating the behavior of system components without changing code nor requiring the cooperation of the components being governed (http://www.policy-workshop.org/). Moreover, they have important analogues as regulatory mechanisms in animal societies and human cultures [15]. Elsewhere we have pointed out the many benefits of policy-based approaches, including reusability, efficiency, extensibility, context-sensitivity, verifiability, support for both simple and sophisticated components, protection from poorly-designed, buggy, or malicious components, and reasoning about component behavior [5].

In this chapter, we describe how policies can be used to represent and help implement adjustable autonomy and mixed-initiative interaction. Previously, several research groups have grappled with the problem of characterizing and developing practical approaches to address these issues. However, each group takes a little different approach and uses variations of the same terminology in a somewhat different fashion. In this chapter, we will briefly characterize a few of the better-known approaches with respect to some common dimensions in order to better understand this important but ill-characterized topic. We will also sketch the approach to implementation we are developing in the context of our research on policy-governed autonomous systems.

As foundation to the remainder of the chapter, section 2 describes our view of the major dimensions of adjustable autonomy; and section 3 does likewise for mixed-initiative interaction. In section 4, we describe our attempts to address these concerns within the KAoS policy and domain management services framework and illustrate how the framework is used within some current applications. Then we briefly discuss related work (section 5). Section 6 provides a brief summary and an outline of next steps.

## 2 Dimensions of Adjustable Autonomy

In this section, we informally describe our view of the dimensions of adjustable autonomy.[1] Section 2.1 briefly discusses the concept of autonomy itself. In section 2.2, we give a description of the major dimensions under consideration, and in section 2.3, we outline basic concepts relating to adjustable autonomy.

### 2.1 Autonomy

No description of adjustable autonomy can proceed without at least some discussion of the concept of autonomy itself. The word, which is straightforwardly derived from a combination of Greek terms signifying self-government (*auto-* (self) + *nomos* (law)).[2] In common usage, we find two subtly different senses of the word autonomy as applied to people or systems acting in the world. 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

---

[1] A formal description of these concepts is currently being developed.

[2] Some excellent detailed analyses of the concept of autonomy that go beyond what can be treated in this chapter have been collected in [21] and in the current volume. We note that subtle differences in the use of the term *autonomy* sometimes affect the slant or emphasis that different researcher put on various aspects of their respective conceptualizations. Note, for example, Brainov and Hexmoor's emphasis on degree of autonomy as a relative measure of independence between an agent and the physical environment, and within and among social groups [8]. Luck *et al.* [28], unsatisfied with defining autonomy as a wholly relative concept, argue that the self-generation of goals should be the defining characteristic of autonomy, thus allowing it to be regarded in absolute terms that more clearly reflect the priority of the aspect of self-sufficiency.

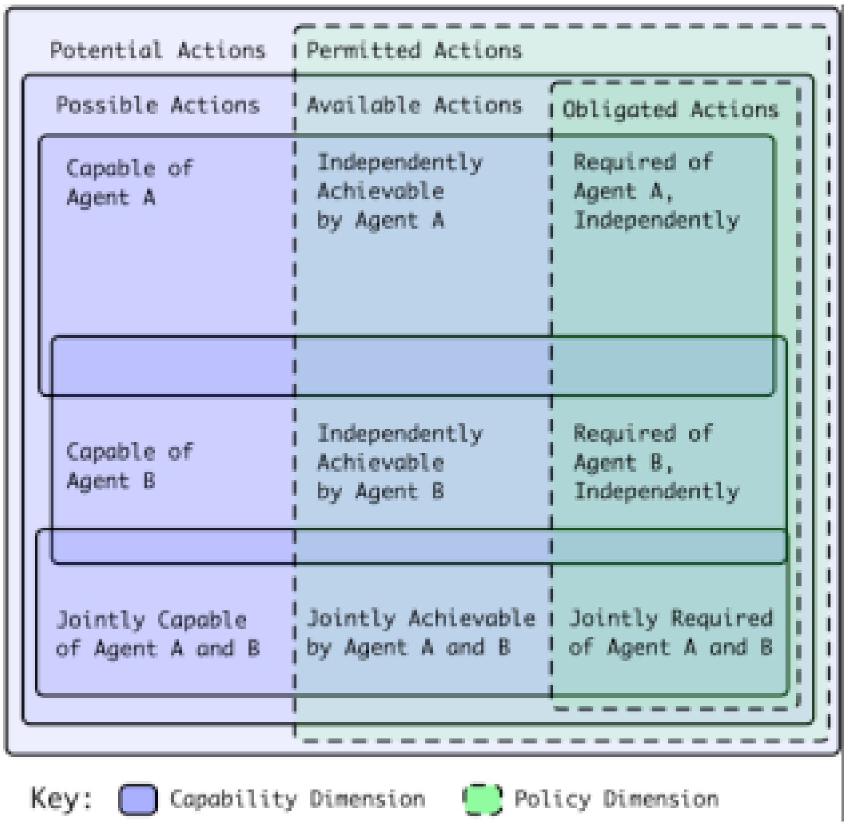


**Figure 1.** Dimensions of autonomy.

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."[3]

### 2.2 Description of the Dimensions

Some important dimensions relating to autonomy can be straightforwardly characterized by reference to figure 1.[4] Note that there are two basic dimensions:

---

[3] 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.

[4] We can make a rough comparison between some of these dimensions and the aspects of autonomy described by Falcone and Castelfranchi [14]. 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-

- 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 some ontology under current consideration.[5] In other words, it contains the union of all actions for all actors currently known to the computational entities that are performing reasoning about adjustable autonomy and mixed-initiative interaction. Note that there is no requirement that all actions that an agent may take be represented in the ontology; only those which are of consequence for policy representation and reasoning need be included.

The rectangle labeled *possible actions* represents the set of potential actions whose achievement by some actor or set of actors is deemed plausible in a given situation. Note that the definition of possibilities is strongly related to the concept of affordances [18; 29], in that it relates the features of the situation to some class of actors capable of exploiting these features in the performance of a given action.[6]

Of these possible actions, any given actor[7] (e.g., Agent A) will likely only be deemed to be *capable of* performing some subset. Capability is a function of the *abilities* and *resources* available to an actor attempting to undertake some action. An actor's ability is the sum of its own knowledge and skills, whereas its resources consist of all other assets it can currently draw on in the performance of actions. Two actors, Agent A and Agent B, may have both overlapping and unique capabilities.[8]

If a set of actors is *jointly capable of* performing some action, it means that it is deemed to be possible for the action to be performed by relying on the capabilities of

---

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.

[5] 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.

[6] As expressed by Norman: "Affordances reflect the possible relationships among actors and objects: they are properties of the world" [31].

[7] For discussion purposes, 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).

[8] Note that figure 1 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 agents in a given situation. For example, although we show A and B sharing the same set of possible actions, this need not always be the case. Also, note that the range of jointly achievable actions has overlap only with Agent B and not Agent A.

both actors. Some actors may be capable of performing a given action either individually or jointly; other actors may not be so capable.

Along the prescriptive dimension, policies specify the various *permissions* and *obligations* of actors [13]. *Authorities* may impose or remove involuntary policy constraints on the actions of actors.[9] Alternatively, actors may voluntarily enter into *agreements* that mutually bind them to some set of policies, so long as the agreement is in effect. The *effectivity* of an individual policy is the set of conditions that determine when it is in or out of force.

The set of *permitted actions* is defined by *authorization policies* that specify which actions an actor is allowed (*positive authorizations* or *A+* policies) or not allowed (*negative authorizations* or *A-* policies) to perform in a given context.[10] The intersection of what is possible and what is permitted to a given set of actors defines a set of *available actions*.

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

Finally, the set of *obligated actions* is defined by *obligation policies* that specify actions that an actor is required to perform (*positive obligations* or *O+* policies) or for which such a requirement is waived (*negative obligations* or *O-* policies). Positive obligations commit the resources of actors, reducing their current overall capability accordingly. *Jointly obligated actions* are those that two or more agents are explicitly required to perform.

Figure 2 contrasts the general case to its extremes.[11] *Absolute freedom,* a condition representing the extreme of independence from deontic constraints governing an actor's actions, is attained when every potential action is permitted, making any action that is possible available to the actor, and any action of which the actor is capable achievable to it. *Absolute capability,* a condition representing the extreme of self-sufficiency, is attained when an actor is capable of any possible action, making any action that is available achievable to it. *Absolute autonomy* combines absolute freedom and absolute capability, meaning that only the impossible is unachievable.

---

[9] Authority relationships may range between those that are static and fixed to those that are determined by negotiation and persuasion as action unfolds.

[10] 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 note that obligations typically require allocation of finite resources; when obligations are no longer in effect, these resources may become free for other purposes.

[11] To simplify the diagram, the dimension of obligation is omitted. Note that absolute capability and absolute autonomy—by definition—require that there are no obligations in effect, since any active obligation reduces current capability in some measure.

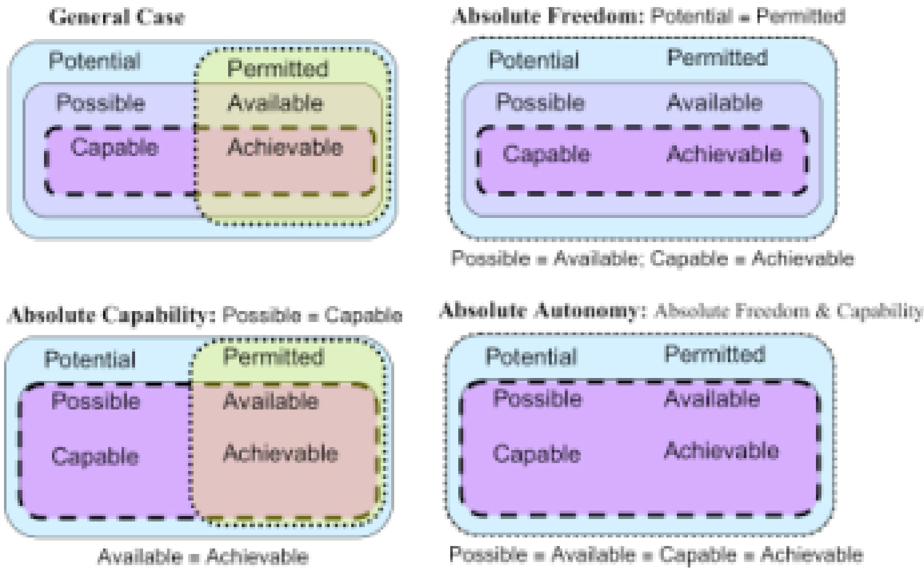


**Figure 2.** The general case and its extremes.

## 2.3 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 whatever performance expectations have been 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 better:[12] 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 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).[13] Assurance that agents will operate safely within well-defined bounds and that they will respond in a timely manner to external control is required for them to be acceptable to people in the performance of non-trivial tasks. People need to feel that agents will handle unexpected circumstances

---

[12] In fact, the multidimensional nature of autonomy argues against even the effort of mapping the concept of "more" and "less" to a single continuum.

[13] 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.

requiring adjustment of their current state of autonomy flexibly and reliably. To the degree adjustable autonomy can be successfully implemented, agents are kept, to the degree possible, from exceeding the limits on autonomy currently in effect, while being otherwise free to act in complete autonomy within those limits. Thus, the coupling of autonomy with adequate autonomy adjustment mechanisms gives the agent maximum opportunity for local adaptation to unforeseen problems and opportunities while assuring humans that agent behavior will be kept within desired bounds.

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 (figure 3). Reducing permissions may be useful when it is concluded, for example, that an agent is habitually attempting actions that exceed its current capabilities—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 performance penalties.

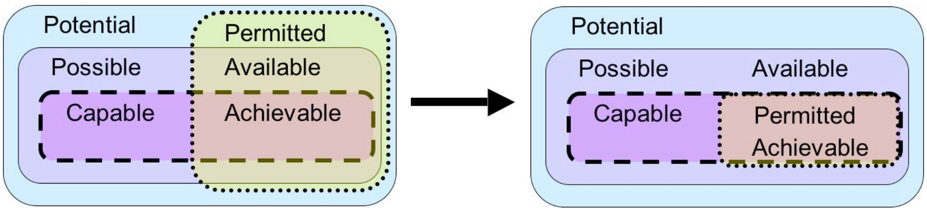


**Figure 3.** Reducing permissions to prevent outstripping capabilities.

If, on the other hand, an agent is known to be capable of actions that go beyond what it is currently permitted to do, its permissions could be increased accordingly (figure 4). 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 [17] [6].

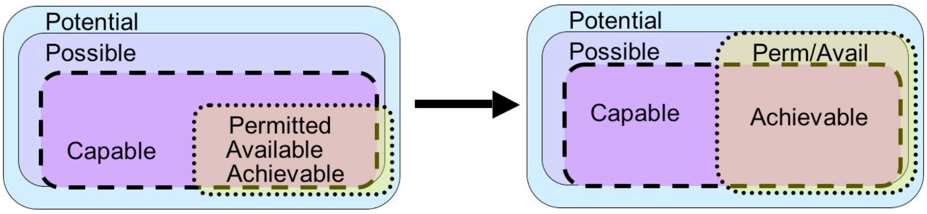


**Figure 4.** Increasing permissions to take full advantage of capabilities.

**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. 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 required to perform any action that outstrips its permissions, capabilities, or possibilities.[14] An "overcommitted" agent can sometimes have its autonomy adjusted to manageable levels through reducing its current set of obligations (figure 5). 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.

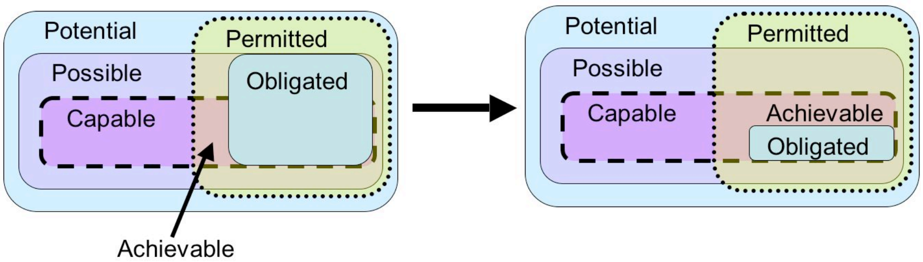


**Figure 5.** Decreasing obligations to match capabilities.

**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.[15]

In some circumstances, it may be possible to adjust autonomy by increasing the set of possibilities available to an agent (figure 6). For example, a mobile agent may

---

[14] 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.

[15] 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.

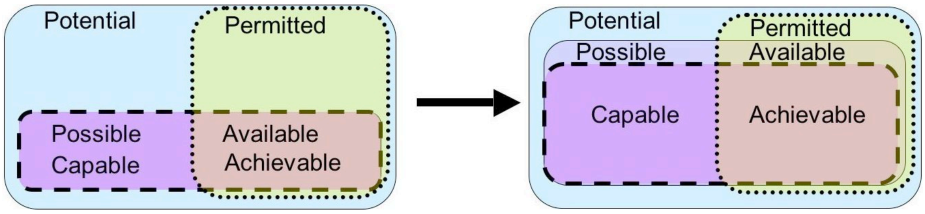


**Figure 6.** Increasing possibilities to leverage unused capabilities.

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 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. An agent's capabilities can also be affected by adding or reducing needed resources.

An adjustable autonomy service aimed at increasing an agent's capabilities (as in figure 7) 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).

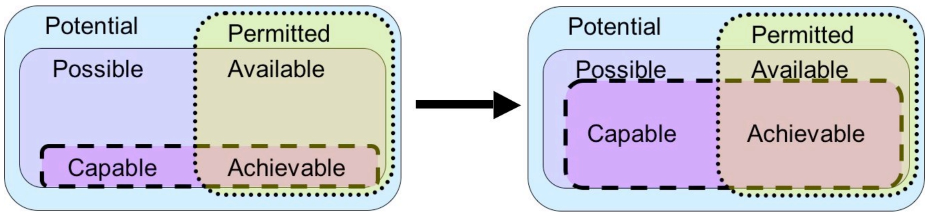


**Figure 7.** Increasing an agent's capabilities to take better advantage of the set of actions available to it.

Having described the principal dimensions of autonomy and the kinds of adjustments that can be made, we now analyze the concept of mixed-initiative interaction from that perspective.[16]

## 3 Mixed-Initiative Interaction

It is enlightening to look at the topic of mixed-initiative interaction in the context of the onward march of automation generally.[17] The concept of automation—which began with the straightforward objective of replacing whenever feasible any task currently performed by a human with a machine that could do the same task better, faster, or cheaper—began to attract the notice of early human factors researchers who began the attempt to systematically characterize the general strengths and weaknesses of humans and machines [16]. The resulting discipline of *function allocation* aimed to provide a rational means of determining which system-level functions should be carried out by humans and which by machines.

Over time it became plain to researchers that things were not as simple as they first appeared. For example, many functions in complex systems are shared by humans and machines; hence the need to consider synergies and conflicts among the various performers of joint actions. Also, the suitability of a particular human or machine to take on a particular task may vary by time and over different situations; hence the need for *dynamic function allocation* [20]. Moreover, it has become clear that automated help of whatever kind does not simply enhance our ability to perform the task: it changes the nature of the task itself [10; 30]. Those who have had a five-year-old child help them by doing the dishes know this to be true—from the point of view of an adult, such "help" does not necessarily diminish the effort involved, it merely effects a transformation of the work from the physical action of washing the dishes to the cognitive task of monitoring the progress (and regress) of the child.

With all these complications, even the pioneers of function allocation have been constrained to admit only limited success in implementing this concept in practice [35]. And so it is that any researcher proposing to design and develop systems manifesting mixed-initiative behavior must approach the task with humility—since such systems will not only manifest all the complexities heretofore mentioned, but also

---

[16] In this chapter we do not address the question of how to evaluate the quality of autonomy adjustment and mixed-initiative interaction. See [12; 19; 23] for a sampling of perspectives on this issue. We note that there are many criteria that can play into such an assessment, including *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*(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), *adaptability* (ability to respond to changes in context), and *task performance* (overall economic and cognitive costs and benefits expressed as utility).

[17] See [3; 10; 20; 22; 32] for insightful perspectives on these and related topics.

aim to delegate the task of dynamic function allocation to the automated elements themselves.

The concept of *mixed-initiative interaction,* involving some combination of humans and/or agents has been succinctly described by Allen as follows:

> "Mixed-initiative refers to a flexible interaction strategy, where each agent can contribute to the task what it does best. Furthermore, in the most general cases, the agents' roles are not determined in advance, but opportunistically negotiated between them as the problem is being solved. At any one time, one agent might have the initiative—controlling the interaction—while the other works to assist it, contributing to the interaction as required. At other times, the roles are reversed, and at other times again the agents might be working independently, assisting each other only when specifically asked. The agents dynamically adapt their interaction style to best address the problem at hand" [1, p. 14].

The following subsections define the concept of mixed-initiative interaction in more detail. We first describe the essential characteristics of joint activity and joint action (3.1). Then we show how the concepts of joint activity and adjustable autonomy come together to enable mixed-initiative interaction (3.2). Finally, we show how policy comes into play within an example (3.3).

## 3.1 Joint Activity and Joint Action[18]

An understanding of joint activity must be at the heart of any formulation of mixed-initiative interaction. Our concept of joint activity relies on the work of Herbert Clark [11], who borrows the following definition from Levinson [27, p. 69]:

> "I take the notion of an activity type to refer to a fuzzy category whose focal members are goal-defined, socially constituted, bounded, events with constraints on participants, setting, and so on, but above all on the kinds of allowable contributions. Paradigm examples would be teaching, a job interview, a jural interrogation, a football game, a task in a workshop, a dinner party, and so on."[19]

Although there are many variations to joint activity, there are several core elements that seem to be common to them all:

---

[18] See Klein and Feltovich [26] for a more complete discussion of the implications of Clark's work for coordination of joint action. We rely heavily on their analysis in this section.

[19] The relationship between language and joint activity is described by Clark as follows: "When people use language, it is generally as part of a joint activity.... The argument is that joint activities are the basic category, and what are called discourses are simply joint activities in which conventional language plays a prominent role. If we take language use to include such communicative acts as eye gaze, iconic gestures, pointing, smiles, and head nods—and we must—then all joint activities rely on language use. Chess may appear to be nonlinguistic, but every chess move is really a communicative act, and every chess game a discourse" [11, p. 58].

- *Intention to produce a genuine, multi-agent product:* The overall joint activity should be aimed at producing something that is a genuine joint project, achieved differently (e.g., faster, better) than any one party, or two parties working alone could do.
- *Interdependency:* It follows that there must be interdependencies among the parties' actions; if the parties' actions have no interdependency, then they are not involved in joint activity (although they may be involved in something that might be thought of as "parallel" activity).
- *Coordination:* There must be coordination with regard to elements of interdependency.
- *Coordination devices:* There must be devices, mutually understood by the parties, that guide coordination. Some of these devices are discussed in more detail below.
- *Common ground:* There must be shared knowledge and interpretation; although the parties' knowledge and interpretations need not be exactly alike, they should have enough commonality to enable the joint activity to move positively in the direction of its goal.
- *Repair:* When there is evidence of loss of common ground—loss of sufficient common understanding to enable joint activity to proceed—there are mechanisms engaged that aim to restore it, to increase common understanding.

Joint activity is a *process*, extended in space and time. There is a time when the parties enter into joint activity and a time when it has ended. These are not "objective" points of time that would necessarily be agreed on by any "observer-in-the-world," but most importantly are interpretations arrived at by the parties involved [11, p. 84]. In some circumstances the entry and exit points may be very clear such as when two people play a classical duet; the same would probably not be said of two musicians doing jazz improvisation.

The overall structure of joint activity is one of embedded sets of actions, some of which may also be joint and some of which may be accomplished more or less individually. All these actions likewise have entry and exit points, although as we have mentioned earlier, these points are not epistemologically "objective." Synchronizing entry and exit points of the many embedded phases involved in complex joint activity is a major challenge to coordination.[20]

So, how does coordination happen? Given a structure of embedded actions—some of which may be joint actions—as well as overall joint activity, this appears to be two questions. How does coordination take place in the more local joint acts that make up an overall joint activity, and how does coordination take place at the more macro level of the overall joint activity itself. With regard to the first, the "coordination devices" [11, pp. 64-65] play a major role:

[20] Clark, in fact, defines joint actions in terms of coordination: "A joint action is one that is carried out by an ensemble of people acting in coordination with each other" [11, p. 3].

- *Agreement:* Coordinating parties are sometimes simply able to communicate their intentions and work out elements of coordination. This category includes diverse forms of signaling that have shared meaning for the participants, including language, signs, gestures, and the like.
- *Convention:* Often, prescriptions of various types apply to how parties interact. These can range from rules and regulations, to less formal codes of appropriate conduct such as norms of practice in a particular professional community, or established practices in a workplace. Coordination by convention depends on structures outside of a particular episode of joint activity.
- *Precedent:* Coordination by precedent is like coordination by convention, except that it applies to norms and expectations developed within an episode of the ongoing process of a joint activity (or across repeated episodes of such activity if the participants are a long-standing team that repeats conduct of some procedure): "That's the way we did it last time."
- *Salience:* Salience is perhaps the coordination device that is most difficult to understand and describe.[21] It has to do with how the ongoing work of the joint activity arranges the workspace so that next move becomes highlighted or otherwise apparent among the many moves that could conceivably be chosen. For example, in a surgery, exposure of a certain element of anatomy, in the course of pursuing a particular surgical goal, can make it clear to all parties involved what to do next. Coordination by salience is a sophisticated kind of coordination produced by the very conduct of the joint activity itself. It requires little or no overt communication and is likely the predominant mode of coordination among long-standing, highly practiced teams.

Coordination across the entire course of an extended joint activity is in some ways similar and in some ways different from the more local coordination involved in individual joint actions (and subactions). For instance, there may be "scripted" conventions for conducting an entire procedure just as there are for conducting more local components of it. That is, joint activities can be more or less open in execution, depending on the presence of applicable norms, policies and the like. In addition to regulatory coordination mechanisms, there are other kinds of macro guides that serve to coordinate across the course of an entire joint activity. Examples are plans for some activity worked out in advance by the participants, or the prior extensive outline worked out by the authors involved in writing a joint academic manuscript. It has been argued that some of the reasons for "standardizing" procedures are to aid coordination and to prevent untoward interactions so that some earlier move does not clobber some necessary later move (e.g., [34]). Of course, any of these overarching coordination devices usually needs to be revisited, and very likely adjusted, as the actual work unfolds.

---

[21] Part of the complication is the relationships among these mechanisms. For example, conventions and precedents may be essential in salience "assignment."

### 3.2 Toward a Better Understanding of Mixed-Initiative Interaction

With this description of joint activity, coupled with the discussion of the dimensions of adjustable autonomy in section 2, we are prepared to better understand mixed initiative interaction. To make this discussion more concrete, we will illustrate with reference to different sorts of vacuum cleaners that embody a spectrum of autonomy.

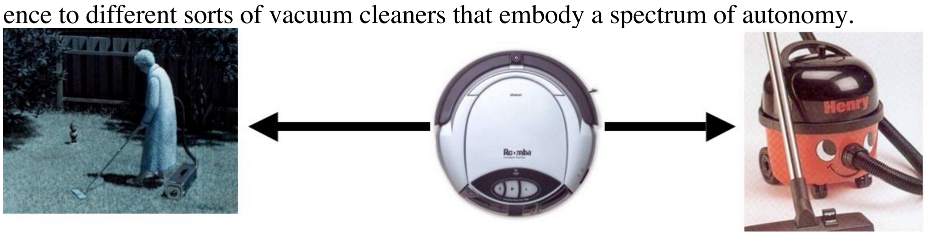


**Figure 8.** Three vacuum cleaners illustrating a spectrum of autonomy.

Representing the "most manual" left end of the spectrum, a plain old vacuum is directly operated as an extension of the woman's arm.[22] Apart from the sweeping and sucking action of the motor, every action is taken at the initiative and direction of the human.

On the "most autonomous" right end of the spectrum, we can imagine a "fully autonomous" vacuum that not only does the actual vacuuming on its own, but also decides when it is time to vacuum and turns itself on, decides when it is time to stop and turns itself off, and even retreats to a closet where it plugs itself into the wall and recharges in anticipation of its next sortie.[23]

We see the process of taking initiative as a particular manifestation of autonomy. In everyday use, the term *initiative* refers to the right, power, or ability to select, begin, execute, or terminate some activity. We speak of doing something "on one's own initiative" as referring to a situation where the individual has relied on his or her own discretion to act independently of outside influence or control.

Mixed-initiative interaction of necessity requires that both parties be somehow involved in directing at least some shared aspects of the joint activity. It is hard to imagine any controversy over the claim that neither of the two extremes represented by the manual and the totally autonomous vacuum qualify as "mixed-initiative interaction"—in the one case the person is taking all the initiative, and in the other the person need take none.

Somewhere between these extremes of human and machine responsibility for the interaction is the basic model of iRobot's Roomba. Its design is fixed such that the

---

[22] Thanks to Ron Francis for permission to use this reproduction of his oil painting entitled "Vacuum Cleaner." Of this painting, Francis writes, "It is easy for someone to get a little lost in this world, and not be noticed. This person is harmlessly vacuuming her back yard. My mother once had a dressing gown just like hers."

[23] We could of course take this to a further extreme where the vacuum not only is responsible to recharge itself, but also takes responsibility for paying its share of the electric bill, hires itself out in its spare time to earn money for the bill, repairs itself, and on ad infinitum to the further reaches of unlimited autonomy.

user must always be the one to take responsibility to switch the vacuum on, tell it how long to run, and put it away and recharge it. Once it is commissioned, the Roomba is always fully responsible for figuring out where to go and what to do until its battery runs low, it completes its work cycle, or the user manually stops it and carts it away.

Although it could be argued that the Roomba is semi-autonomous, it is our view that its interaction with the user could not be classed as mixed-initiative. True it is that each party has a reasonable degree of autonomy from the other. It is also obvious that the action of vacuuming could have only been achieved through the participation of both parties, each party having contributed something unique to the task. What is missing, however, is the chance for either party to determine opportunistically who should perform which tasks and when they should be done. The tasks that the human does are tasks that only the human can do, the actions the vacuum takes cannot be performed by the human,[24] and these respective roles and responsibilities are fixed in advance and unalterable.

In short, we can say that necessary conditions for mixed initiative interaction are:

- Engagement in a joint activity (including some manifestation of the kinds of properties of joint activity described by Clark);
- At least some aspects (e.g., proposing, deciding, initiating, executing, monitoring, terminating) of some actions supporting the joint activity can be achieved individually by two or more of the participants;
- No set of policies uniquely fixes who must perform all aspects of these actions, or when they must be performed; thus any one of the participants capable of doing so is permitted to take initiative as circumstances unfold.[25]

### 3.3 The Role of Policy in Mixed-Initiative Interaction

To understand the role that policy could play in mixed-initiative interaction, we extend with the previous example to include a hypothetical mixed-initiative vacuum cleaner (figure 9). Let's assume for starters that there is an overall activity "Clean living room," which is something that can only be achieved by the human and vacuum working jointly. The action of "Turn on/off," in our example, is something that is achievable by the human alone; and the "vacuum" action is something that is achievable by the vacuum alone.

There are three nodes in the tree that are identified as potential mixed-initiative actions, i.e., that can potentially be accomplished through mixed-initiative interaction. "Select location" is something that could either be done by both parties jointly, or by either one alone; we assume that "move to" and "empty bag" are actions that could be

[24] One could argue that in some sense the human can clumsily take the initiative with respect to some of the actions normally performed by the vacuum, e.g., in determining where the vacuum should move by lifting it up and carrying it elsewhere.

[25] And in fact one of the participants *must* take initiative for the action to proceed.

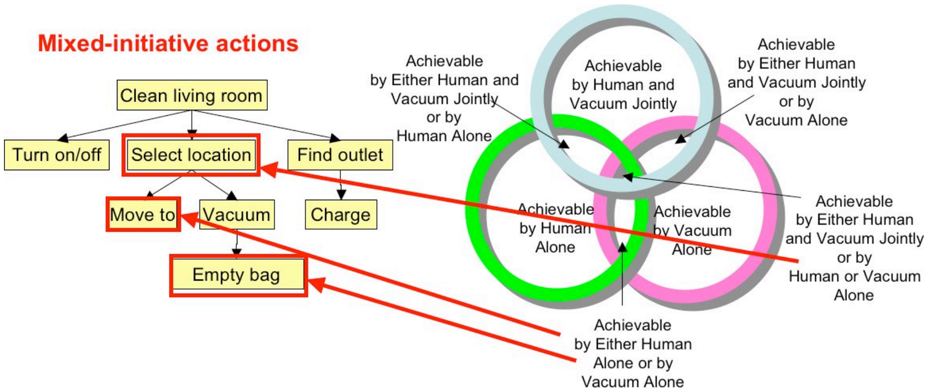


**Figure 9.** Opportunities for mixed-initiative interaction with a vacuum cleaner.

achieved by either the human alone or the vacuum alone but not both working together.

Recalling Clark's adoption of Levinson's definition of joint activity, we can see the role of policy in representing "focal members [who] are goal-defined, socially constituted, bounded, events with constraints on participants, setting, and so on, but above all on the kinds of allowable contributions." As the participants begin to engage in joint activity, they bring with them a history of agreements, conventions, precedents, and salience-related expectations that serve to coordinate their joint actions.

Although, thanks to lifelong experience, most humans come pre-packaged with a host of ready-made conventions, expectations, and the like that cover everyday situations, some kind of representation of these sorts of conventions and expectations needs to be explicitly "added in" to artificial systems in order to help them work well with people. This is a different concept of automation than the one that has previously been the basis for the design of generations of "strong, silent" systems that permit only two modes: fully automatic and fully manual [10]. In practice the use of such systems often leads to situations of human "underload," with the human having very little to do when things are going along as planned, followed by situations of human "overload," when extreme demands may be placed on the human in the case of agent failure.

In contrast, within systems capable of mixed-initiative interaction, policies and other relevant information needed for coordination could be explicitly represented within the artificial system in some form of "agreement" intended to govern selected aspects of joint activity among the parties. While this should not be mistaken as a requirement for full-blown anthropomorphism, it is clear that at least some rough analogues to human coordination mechanisms will be required in order to assure effective teamwork among people and agents.

With reference to our hypothetical mixed-initiative vacuum, we now describe examples of policies from five categories:

- Policies affecting initiative,
- Policies affecting delegation,
- Notification policies,

- Supervisory policies, and
- Policies constraining human actions.

These categories and examples are intended to be illustrative, not comprehensive.

**Policies affecting initiative.** Note that the heading of this subsection refers to "policies affecting initiative" and *not* "policies determining initiative." Of necessity, the decision about whether and when to take initiative relative to a particular joint action is determined by the agent consulting its own reasoning processes, and not by the policy-related components. Policies, however, can affect the process of initiative-taking in a number of ways, or even terminate it altogether. For example, an event may trigger an obligation policy that uniquely requires one or the other of the parties to initiate some aspect of the joint action, thus foreclosing future opportunities for mixed-initiative interaction. For example, consider the following positive obligation policy:

`O+: If the vacuum finds a baby on the floor, then the human must immediately tell the vacuum where to move to next.`

Once the obligation is in effect, the decision has been made about where the vacuum moves to next, and the possibility of further mixed-initiative interaction on that decision is gone. If, however, the human happens to possess a new advanced model of the vacuum cleaner that is baby-safe, this requirement may be waived by a negative obligation policy of the following sort:

`O-: If the vacuum in question is a super-duper model, it is not obliged to have the human tell it where to move to next when a baby is found on the floor.`

Other obligations may require the termination of some joint action by one or the other of the actors:

`O+: If an actor determines that the joint action has been achieved, has become unachievable, or has become irrelevant, it is required to terminate its efforts with regard to the joint action and to notify the other parties.`

There may also be policies of other sorts relating to initiative. For example, policies might be specified that not only affect what the parties can do and when they can do it, but also constrain which parties of a joint action can decide what they can do and when they can do it, or how they need to negotiate about who's going to decide these things.

**Policies affecting delegation.** Authorization policies may be needed in some scenarios—for example, to allow actors to take the initiative in delegating to other actors in teamwork scenarios:

`A+: The vacuum is permitted to delegate vacuuming an area to any other vacuum of the same make and model (or better) in the room.`

Or:

`A-: The vacuum is forbidden from delegating vacuuming tasks to the toaster.`

Note that delegation is handled as just another kind of action that may or may not be authorized or obliged. Actions that add, modify, or delete policies can be constrained in a similar way without resorting to special "meta-level" mechanisms.

**Notification policies.** The fact that who will take the initiative on various aspects of joint actions is not determined in advance means that there must be adequate means for each party to determine the other parties' state and intentions and coordinate its own actions accordingly.[26] Obligation policies can be used, for example, to make sure that the vacuum reports relevant aspects of its state and intentions at appropriate times:

`O+: The vacuum must notify the human about the state of its battery, its position, and the estimated time remaining to finish the room every five minutes.`

Obligation policies can also be used to avoid conflicts or duplication of effort:

`O+: If the human tries to empty the bag while the vacuum is already trying to empty the bag, then signal the human.`

**Supervisory policies.** Humans will not expect even the most competent vacuum to be trusted to do the right thing in every situation. This fact motivates the requirement for policies that guard against certain actions taking place:

`A-: The vacuum is not permitted to operate after midnight.`

Similar considerations motivate obligations that require the vacuum to gain approval before proceeding with certain actions:

`O+: The vacuum must obtain permission from the user before entering a new room.`

An appropriate negative authorization policy tied to this policy could prevent the vacuum from performing this action (or perhaps any other action) until human approval had been obtained.

Policies allowing the vacuum to take initiative in executing fallback behavior when the human supervisor is not available may also be important:

`A+: The vacuum is permitted to enter a new room (just this once) if permission from the human has been requested more than ten minutes ago and the human has not responded.`

**Policies constraining human actions.** Of course, there may also be situations where trust of the human operator is limited.[27] This fact motivates the requirement to be able to define policies that constrain human actions:

---

[26] Given some requirement for notification, there is also a role for context-sensitive policies and personal preferences regarding the salience, latency, and mode of notification. How one might factor in such considerations is discussed in [9; 33].

[27] We note that if the vacuum is preventing the human from operating it dangerously, it is really ultimately the authority of the administrator who defined the policy who is preventing the operator, not the vacuum itself.

```
O+: The vacuum must prevent the human from deliberately
crashing it into an obstacle when its movements are un-
der manual control.
A-: Unauthorized humans are forbidden from telling the
vacuum where to move.
```

In research funded by the Office of Naval Research, DARPA, the Army Research Labs, and NASA, we are currently conducting research to develop and evaluate policies that will facilitate mixed-initiative interaction in conjunction with a testbed that integrates the various capabilities of TRIPS, Brahms, and KAoS [4]. The next section gives an overview of KAoS, and describes how it is being used to support adjustable autonomy and mixed-initiative interaction.

## 4 KAoS

KAoS a collection of componentized policy and domain management services compatible with several popular agent frameworks, including Nomads [36], the DARPA CoABS Grid [25], the DARPA ALP/Ultra*Log Cougaar framework (http://www.cougaar.net), CORBA (http://www.omg.org), Voyager (http://www.recursionsw.com/osi.asp), Brahms (www.agentisolutions.com), TRIPS [2; 4], and (soon) 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 [24; 41]. KAoS has been deployed in a wide variety of applications, from coalition warfare [7; 37] and agile sensor feeds [38], to process monitoring and notification [9], to robustness and survivability for distributed systems [http://www.ultralog.net], to semantic web services composition [41], to human-agent teamwork in space applications [6], to cognitive prostheses for augmented cognition [5].

*KAoS domain services* provide the capability for groups of software components, people, resources, and other entities to be organized into domains and subdomains to facilitate agent-agent collaboration and external policy administration.

*KAoS policy services* allow for the specification, management, conflict resolution, and enforcement of policies within domains.

### 4.1 KAoS Policy Ontologies

The current version of the core KAoS Ontologies (http://ontology.ihmc.us/) defines basic concepts for actions, actors, groups, places, various entities related to actions (e.g., computing resources), and policies. It includes more than 100 classes and 60 properties.

The core *actor ontology* contains classes of people and software components that can be the subject of policy. Groups of actors or other entities may be distinguished according to whether the set of members is defined extensionally (i.e., through explicit enumeration in some kind of registry) or intentionally (i.e., by virtue of some common property such as types of credentials actors possess, or a given place where various entities may be currently located).

The core *action ontology* defines various types of basic actions such as accessing, communication, monitoring, moving, and so forth. An ontological definition of an action associates with it a list of properties describing context of this action or a current state of the system relevant to this action. Example properties of action classes are, for instance: destination of the communication, type of encryption used, resources accessed, time, previous history, and so forth. Each property is associated with the definition of a range of values it could have for each of the action classes. A particular instance of the action class can take values on the given property only from within this range. Actions are also divided into *ordinary actions* and *policy actions,* the latter comprising those actions that have to do with the operations of the KAoS services themselves[28].

For a given application, the core KAoS ontologies are usually further extended with additional classes, individuals, and rules, which use the concepts defined in the core ontologies as superconcepts. This allows the framework to discover specialized concepts by querying an ontology repository for subclasses or subproperties of the given concept or property from the core ontologies. For example additional application-related context could be added to actions such as specific credentials used in a given environment.

During the initialization process, the core policy ontologies are loaded into the *KAoS Directory Service* using the namespace management capabilities of the *KAoS Policy Administration Tool (KPAT)* graphical user interface. Additional application-specific or platform-specific ontologies can then be loaded dynamically using KPAT or programmatically using the appropriate Java method. A distributed version of the KAoS Directory Service is currently being implemented. We are also studying possibilities for interaction among multiple instances of Policy Services [41].

The Directory Service is also informed about the structure of policies, domains, actors, and other application entities. This information is added to the ontology repository as instances of concepts defined in pre-loaded ontologies or values of these instance properties. As the end-user application executes, instances relating to application entities are added and deleted as appropriate.

KAoS employs the Jena Semantic Web Toolkit by HP Labs in Bristol (http://www.hpl.hp.com/semweb) to incrementally build OWL definitions and to assert them into the ontology repository managed by the Directory Service. In order to provide description logic reasoning on the OWL defined ontologies, the Java Theorem Prover (http://www.ksl.stanford.edu/software/JTP) inference engine has been integrated with KAoS. Performance is always an issue in logic reasoning; however, the steady

[28] This distinction allows reasoning about actions on policies and the policy framework without resorting to the use of special "metapolicy" mechanisms.

improvement of JTP has led to a dramatic increase in its performance—an order of magnitude or more in some cases—in the last two years. The most time consuming operation in JTP is asserting new information, which happens mostly during system bootstrap. Currently, loading of the KAoS core ontologies takes less than 16 seconds on Pentium III 1.20 GHz with 640 MB RAM. Adding a policy takes usually less than 340ms. Querying JTP about ontology concepts and policies is much faster and takes only a few milliseconds.

### 4.2 Policy representation

In KAoS, policies can express *authorization* (i.e., constraints that permit or forbid some action) or *obligation* (i.e., constraints that require some action to be performed, or else serve to waive such a requirement) for some type of action performed by one or more actors in some situation [5]. Whether or not a policy is currently applicable may be conditional upon some aspect of the situation. Auxiliary information may be associated with a policy, such as a rationale for its existence or a specification of some penalty for policy violation. In contrast to many existing policy systems [http://www.policy-workshop.org], KAoS aims at supporting both an extensible vocabulary describing concepts of the controlled environment and also an evolution of its policy syntax. Such features are one beneficial consequence of defining policies within ontologies and using an extensible framework architecture [39].

In KAoS, a policy is represented as an ontology instance[29] of one of the four types of policy classes: positive or negative authorization, and positive or negative obligation. The instance possesses values for various management-related properties (e.g., priority, time stamp, site of enforcement) that determine how the given policy is handled within the system. The most important property value is the name of a controlled action class, which is used to determine the actual meaning of the policy. Authorization policies use it to specify the action being authorized or forbidden. Obligation policies use it to specify the action being obliged or waived. Additionally the controlled action class contains a trigger value that creates the obligation, which is also a name of the appropriate class of actions. Policy penalty properties contain a value that corresponds to a class of actions to be taken following a policy violation.

---

[29] See *http://ontology.ihmc.us/SemanticServices/S-F/Example/* for an example of KAoS policy syntax.

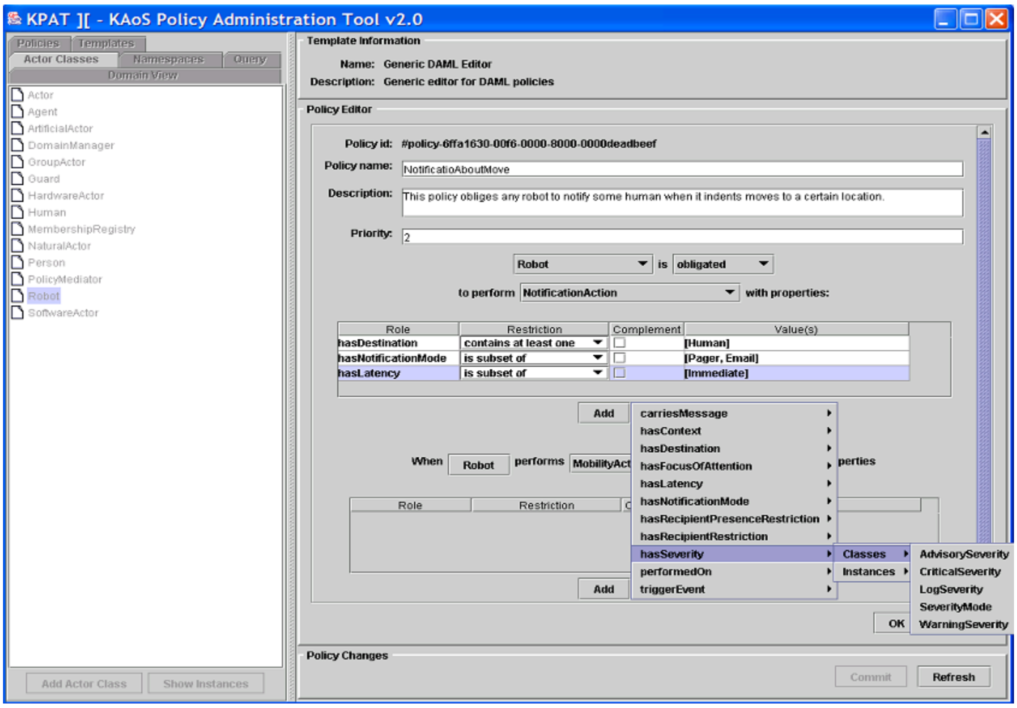


**Figure 10** KAoS Policy Administration Tool (KPAT) policy builder interface.

As seen from this description, the concept of action is central to the definition of KAoS Policy. Typically any action classes required to support a new policy are generated automatically by KAoS when a user defines new policy (usually using KPAT). Through various property restrictions, a given subject of the action can be variously scoped, for example, either to individual agents, to agents of a given class or to agents belonging to a particular group, and so forth. The specific contexts in which the policy constraint applies can be precisely described by restricting values of the action's properties, for instance requiring that a given action be signed using an algorithm from the specified group.

Policy Management

**A** strength of KAoS is in its extensive support for policy life-cycle management. KAoS hides many elements of complexity of this process from the user. KAoS also provides a sophisticated policy disclosure interface enabling querying about policy impact on planned or executed actions.

### 4.3 Graphical interface to ontology concepts

The KPAT graphical interface to policy management hides the complexity of the OWL representation from users. The reasoning and representation capabilities of OWL are used to full advantage to make the process as simple as possible. Whenever a user has to provide an input is always presented with a complete set of values he can choose from, which are valid in the given context.

As in the case of the generic policy editor shown on figure 10, a user, after selecting an actor for a new policy, is first presented with the list of actions the given type of actors is capable to perform based on the definition in the ontology relating actions to actors by the *performedBy* property. When the user selects a particular action type information about all the properties, which can be associated with the given actions, are presented. For each of the properties, the range of possible values is obtained; instances and classes falling into this range are gathered if the user wants to build a restriction on the given property, thus narrowing the action class used in the build policy to its context.

### 4.4 Policy administration

Each time a new policy is added or an existing one is deleted or modified, the potential impact goes beyond the single policy change. Policy administrators need to be able to understand such interactions and make sure that any unwanted side effects are eliminated. KAoS assists administrators by identifying instances of given types of policy interactions, visualizing them, and, if desired, facilitating any necessary modifications.

One important type of interaction is a policy conflict [7; 42]. For example, one policy might authorize actor A to communicate with any actor in group B while a new policy might forbid actor A from communicating with actor B1, a member of B. In general, if a new policy overlaps in key properties of a subset of controlled actions with an existing policy of a potentially conflicting modality (i.e., positive vs. negative authorization (as in our example); positive vs. negative obligation; positive obligation vs. negative authorization), some means must be used to identify the conflict and to determine, in the area of overlap, which policy takes precedence[30]. If precedence cannot be determined otherwise, KAoS will ask the administrator to determine the appropriate action [40].

### 4.5 Policy exploration and disclosure

A human user or software component uses KAoS to investigate how policies affect actions in the environment. In general, the answers to these queries are decided by inferring whether some concrete action falls into a category of action controlled by one or more policies, and then determining what conclusions about the described action can be drawn [40].

---

[30] If desired, precedence relations can be predefined in the ontology, permitting partially or totally automated conflict resolution.

### 4.6 Adapting policy to legacy systems

When policy leaves the Directory Service, for performance reasons it typically has to map OWL into a format that is compatible with the legacy system with which it is being integrated. KAoS communicates information from OWL to the outside world by mapping ontology properties to the name of the class defining its range as well to a list with cached instances of that class that were in existence when the policy left the Directory Service. A particular system can use the cached instance for its computation; also in any moment it can refresh the list by contacting the Directory Service and providing the name of the range. Alternatively, the Directory Service can push changes to the system as they occur.

### 4.7 Adjustable autonomy and mixed-initiative interaction

## 5 Related Work

## 6 Summary and Next Steps

Whereas most ontologies involved with agent autonomy are concerned with generating plans for what an agent should *do*, KAoS and its ontologies are one of the few that aim to specify how agent behavior should be *constrained*. Regarding the usefulness of this perspective, Sheridan observes:

"In democracies specification of 'shoulds' is frowned upon as an abridgement of freedom, and bills of basic rights such as that of the United States clearly state that 'there shall be no law against…', in other words declarations of unconstrained behavior. In such safety-sensitive industries as aviation and nuclear power, regulators are careful to make very specific constraint specifications but then assert that those being regulated are free to comply in any manner they choose.

Vicente and Pejtersen assert that constraint-based analysis accommodates much better to variability in human behavior and environmental circumstance. They make the point that navigating with a map is much more robust to disturbance and confusion over detail than navigating with a sequence of directions" [35, pp. 212-213].

Over the next several months we hope to complete the development of a formal model to describe how a combination of ontology-based inference and decision-theoretic methods (informed by empirical observation) can lead to effective autonomy adjustments. We expect many interesting results form the continuation of these studies of the "other side" ** of autonomy.

## References


[1] Allen, J. F. (1999). Mixed-initiative interaction. *IEEE Intelligent Systems*, September-October, 14-16.

[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] Billings, C. E. (1996). *Aviation automation: The Search for a Human-Centered Approach*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[4] 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,

[5] Bradshaw, J. M., Beautement, P., Raj, A., Johnson, M., Kulkarni, S., & Suri, N. (2003). 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. in press). Berlin: Springer Verlag.

[6] 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.

[7] Bradshaw, J. M., Uszok, A., Jeffers, R., Suri, N., Hayes, P., Burstein, M. H., Acquisti, A., Benyo, B., Breedy, M. R., Carvalho, M., Diller, D., Johnson, M., Kulkarni, S., Lott, J., Sierhuis, M., & Van Hoof, R. (2003). Representation and reasoning for DAML-based policy and domain services in KAoS and Nomads. *Proceedings of the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2003)*. Melbourne, Australia, New York, NY: ACM Press,

[8] Brainov, S., & Hexmoor, H. (2002). Quantifying autonomy. In H. Hexmoor, C. Castelfranchi, & R. Falcone (Ed.), *Agent Autonomy*. (pp. 43-56). Dordrecht, The Netherlands: Kluwer.

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

[10] Christofferson, K., & Woods, D. D. (2002). How to make automated systems team players. In E. Salas (Ed.), *Advances in Human Performance and Cognitive Engineering Research, Vol. 2*. JAI Press, Elsevier.

[11] Clark, H. H. (1996). *Using Language*. Cambridge, UK: Cambridge University Press.

[12] 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.

[13] 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.

[14] 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.

[15] Feltovich, P., Bradshaw, J. M., Jeffers, R., & Uszok, A. (2003). Social order and adaptability in animal, human, and agent communities. *Proceedings of the Fourth International Workshop on Engineering Societies in the Agents World,* (pp. 73-85). Imperial College, London,

[16] Fitts, P. M. (Ed.). (1951). *Human Engineering for an Effective Air Navigation and Traffic Control System*. Washington, D.C.: National Research Council.

[17] 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,

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

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

[20] 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.

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

[22] Hoffman, R., Feltovich, P., Ford, K. M., Woods, D. D., Klein, G., & Feltovich, A. (2002). A rose by any other name… would probably be given an acronym. *IEEE Intelligent Systems*, July-August, 72-80.

[23] 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,

[24] 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,

[25] 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,

[26] Klein, G., & Feltovich, P. J. (in preparation). Multiple metaphors for complex coordination. In

[27] Levinson, S. C. (1992). Activity types and language. In P. Drew & J. Heritage (Ed.), *Talk at Work*. (pp. 66-100). Cambridge, England: Cambridge University Press.
[28] Luck, M., D'Inverno, M., & Munroe, S. (2002). Autonomy: Variable and generative. In H. Hexmoor, C. Castelfranchi, & R. Falcone (Ed.), *Agent Autonomy*. (pp. 9-22). Dordrecht, The Netherlands: Kluwer.
[29] Norman, D. A. (1988). *The Psychology of Everyday Things*. New York: Basic Books.
[30] Norman, D. A. (1992). Cognitive artifacts. In J. M. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface*. (pp. 17-38). Cambridge: Cambridge University Press.
[31] Norman, D. A. (1999). Affordance, conventions, and design. *Interactions*, May, 38-43.
[32] Sarter, N., Woods, D. D., & Billings, C. E. (1997). Automation surprises. In G. Salvendy (Ed.), *Handbook of Human factors/Ergonomics, 2nd Edition*. New York, NY: John Wiley.
[33] Schreckenghost, D., Martin, C., & Thronesbery, C. (2003). Specifying organizational policies and individual preferences for human-software interaction. *Submitted for publication*.
[34] Shalin, V. L., Geddes, N. D., Bertram, D., Szczepkowski, M. A., & DuBois, D. (1997). Expertise in dynamic, physical task domains. In P. J. Feltovich, K. M. Ford, & R. R. Hoffman (Ed.), *Expertise in Context: Human and Machine*. (pp. 195-217). Menlo Park, CA: AAAI/MIT Press.
[35] Sheridan, T. B. (2000). Function allocation: algorithm, alchemy or apostasy? *International Journal of Human-Computer Studies*, 52(2), 203-216.
[36] 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,
[37] Suri, N., Bradshaw, J. M., Burstein, M. H., Uszok, A., Benyo, B., Breedy, M. R., Carvalho, M., Diller, D., Groth, P. T., Jeffers, R., Johnson, M., Kulkarni, S., & Lott, J. (2003). DAML-based policy enforcement for semantic data transformation and filtering in multi-agent systems. *Proceedings of the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2003)*. Melbourne, Australia, New York, NY: ACM Press,
[38] 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*.
[39] 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.

[40] Uszok, A., Bradshaw, J. M., & Jeffers, R. (2004). KAoS: A policy and domain services framework for grid computing and semantic web services. *Proceedings of the Second International Conference on Trust Management*. Oxford, England,

[41] 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,

[42] 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,