Artificial Intelligence for Human-Robot Interaction
Papers from the AAAI 2015 Fall Symposium

A Taxonomy for Improving Dialog between Autonomous Agent Developers and Human-Machine Interface Designers

Daylond J. Hooper[1], Jeffrey P. Duffy[1], Gloria L. Calhoun[2], Thomas C. Hughes[1]
[1]Infoscitex, Inc, 4027 Colonel Glenn Hwy, Beavercreek OH 45431
[2]711 HPW/RHCI, 2210 8th St. Bldg. 146, Rm. 122, WPAFB OH 45433
{daylond.hooper.1.ctr, jeffrey.duffy.1.ctr, gloria.calhoun, thomas.hughes.13.ctr}@us.af.mil

Abstract

Autonomous agents require interfaces to define their interactions with humans. The coupling between agents and humans is often limited, with disjoint goals between the agent interface and its associated autonomous components. This leads to a gap in human interaction relative to agent capabilities. We seek to aid interface designs by clarifying agent capabilities within an interface context. A taxonomy was developed that can help elucidate the agent's affordances and constraints that guide interface design. Moreover, the descriptors employed in the taxonomy can serve as a common language to support dialog between agent and interface developers, resulting in improved autonomous systems that support human-autonomy coordination.

Introduction

Key limitations in the strength and usability of a specific human-machine interface (HMI) can be addressed through an improved dialog between agent developers and HMI designers. Communication between the two design communities is critical to developing systems that support appropriate levels of human-autonomy coordination. To improve the dialog, it is of benefit to consider establishing a common language with which HMI designers and agent developers can communicate certain goals and features. For the HMI designers, the language's descriptors communicate desired agent capabilities. The same descriptors can help scope agent development, such that the agent's capabilities are tuned to support interface features. Thus, a common language/taxonomy that aids productive dialog between agent developers and HMI designers helps to address both problems: enabling HMI designers to construct more effective interfaces and scoping feature sets for agent developers to support.

Through an Air Force Research Laboratory research initiative, we considered a number of agent development techniques in the context of designing more effective HMIs. The long-term goal of this research initiative is to support system developments such that future complex Air Force operations benefit from the joint capabilities of advanced agent and HMI technologies. In doing so, we considered the various affordances and constraints of a number of techniques. In evaluating these, we found three key issues: 1) terms used to describe agents by various communities: HMI, agent development, and others (such as from the Department of Defense) rarely overlap; 2) when the terms did overlap, there was a distinct mismatch in the meaning of the terms between the communities; and 3) the lack of joint understanding can cause a disconnect between HMI designers and the developers of the new technologies for autonomous systems. The lack of a common language to describe desired goals and capabilities often leads to limitations in the resulting system's capabilities. Even with feedback between the developers and HMI designers, HMI designers may make feature requests that are considered difficult or infeasible by the agent developers. The absence of a common taxonomy can also affect agent development, as the agent developers, unmindful of the potential HMI concerns or desired capabilities, may develop products that do not support many usability-related features. The HMI in some systems can be unmanageably complex if designed only from the perspective of the autonomous agent system rather than the joint operator-autonomous agent system (Bartram and Ovans 1995), (Steinfeld 2004).

In an effort to alleviate the disconnect between agent descriptions and the desired features of the human interface, we focused on terms that are familiar to HMI designers as they relate to an agent's capabilities. However, even

within the human factors and cognitive psychology literature, the same terms are employed to describe different agent features in the context of human interaction and understanding. There is very little consistency between publications describing desired autonomous agent features (e.g., (Air Force Research Laboratory 2013), (US Air Force 2011), (Department of Defense 2012)), so in order to establish a common language, consistent definitions are needed (Burke et al. 2004), (Korsah, Stentz, and Dias 2013).

This paper reports the results of our efforts to bridge this communication gap between the agent and HMI communities. First, we provide a review of other taxonomies. Then, we introduce descriptors and provide discernable definitions for each. These descriptors are the basis of our proposed taxonomy. Next, we define levels within each of the defined taxonomic descriptors that provide a more granular description, supporting better communications between agent and HMI communities. Finally, we apply the proposed taxonomy to example systems to show how the descriptors help to close the loop between HMI designers and agent developers so that their joint developments can enhance human-autonomy coordination.

Background

Taxonomies for describing agent features, compositions, and interactions with external sources have been around for some time. The taxonomy that we propose in this paper does not preclude any of the taxonomies we have reviewed. Rather, our proposed approach establishes a common language to support dialog between agent and HMI communities that could actually lead to the use of other taxonomies (e.g., those related to agents, human robot interfaces (HRI), human computer interfaces (HCI), interface mechanisms, etc.). The following provides a brief review of some of these as well as sources that guided the design of our taxonomy.

There have been taxonomies created for specific aspects of interactions between humans and robots/agents. On the agent side, there are many taxonomies created for multi-agent systems. (Dudek et al. 1996) created a taxonomy that classifies multi-agent systems according to communication, computational, and other capabilities. (Gerkey and Mataric 2004) created a domain-independent taxonomy for the multi-resource independent task allocation problem in multi-robot systems. This taxonomy was extended by (Korsah, Stentz, and Dias 2013) to address problems with interrelated tasks and constraints. (Farinelli, Iocchi, and Nardi 2004) developed a taxonomy for describing coordination between teams, their relationship/knowledge, and their architectural makeup. A taxonomy to guide the design of multi-agent interaction based on the actions of an agent and their relationship or effects on the rest of the agents is described in (Van Dyke Parunak et al. 2004). The above taxonomies can be used to describe and visualize a team of agents as one collective agent (Humphrey, Gordon, and Adams 2006).

In contrast, there are taxonomies that describe the actual interfacing mechanisms between humans and machines, such as robots. (Seneler, Basoglu, and Daim 2008) describes a taxonomy for characteristics of interfaces that focuses on how the interface is used by the operator rather than the interface-agent relationship. Here the emphasis is on user acceptance of the interface technology, rather than optimizing the interface in terms of the operator's interaction with the agent.

The connecting mechanisms between the agents and the physical interfaces are the crux of many HRI/HCI/HMI taxonomies. For instance, (Yanco and Drury 2002), (Yanco and Drury 2004) created a taxonomy for HRI, which has a primary focus on levels of interaction that describe the architecture and information flow of the human-robot team. They use Sholtz's identified five human roles that contribute to the effectiveness of HRI (Scholtz 2003) as a factor in their taxonomy. These roles dictate the situational awareness needs of the human, which relate to the HMI aspects of our proposed taxonomy. (Bartram and Ovans 1995) also define an information taxonomy, describing communication flow through the interfaces, as opposed to the functionality that the interfaces provide with respect to agent capabilities and operator needs. This taxonomy addresses the supervisory control tasks of monitoring and controlling the autonomous agents.

Aside from these taxonomies, there have been metrics, principles, and lessons reported for HRI systems, further establishing a need for a common language to utilize and discuss these topics. (Steinfeld et al. 2006) established metrics to evaluate and compare implemented HRI systems. In an earlier publication, (Steinfeld 2004) recommended seven topics that those producing interfaces for autonomous systems should address, six of which can be used in describing the human-machine system within our proposed taxonomy. Principles for implementing interfaces and measuring human/robot interaction were also published in (Goodrich and Olsen 2003).

Agent/HMI Focused Taxonomy

The taxonomies and other approaches just described focus on total systems in the context of their applications. Our taxonomic approach focuses instead on the system's agents with the goal of clearly defining relevant tradeoffs that guide HMI design. Even though our approach emphasizes optimizing agent/human operator interaction, our descriptors can also be used to characterize functionality at

the system level. Similarly, the various taxonomies, principles, lessons, and metrics described earlier can help describe lower-level aspects of our taxonomy, informing what is required by both agent and operator team members to optimize interaction.

For our taxonomy, the focus to date has been on defining the descriptors and describing corresponding levels for each. Similarly, (Korsah, Stentz, and Dias 2013) employed a two-level taxonomy by defining terminology that distinguished between task types and applying Gerkey's taxonomy (Gerkey and Mataric 2004) to describe their levels in more detail. Our approach, however, aims to help HMI designers communicate the operator's needs for interacting with the autonomous agent, as well as assisting autonomous agent developers in communicating the agent's capabilities and how it can (or cannot) support certain HMI functionality. This taxonomy is necessary to support agent/HMI developments for autonomous systems.

Descriptors

Our proposed taxonomy is composed of four descriptors: Agility, Directability, Observability, and Transparency. Each descriptor is decomposed into five levels: *None*, *Minimal*, *Limited*, *Moderate*, and *Complete* (the latter is *High* in the case of Agility). These taxonomic levels are described in more detail later. In this section, we provide a working definition for each descriptor, as well as supporting rationale.

Agility

The Air Force Research Laboratory Autonomy S&T Strategy states that "systems will have the robustness and flexibility to assure operations in complex, contested, and dynamic environments" (Air Force Research Laboratory 2013). The terms 'robustness' and 'flexibility' are used to describe this capability. For our definition of agility, we view robustness and flexibility as characteristics that may contribute to agility, since neither term is sufficient for defining agility alone. We define agility:

> ***Agility****–the ability of an agent to respond in an effective manner to new inputs within a short timeframe.*

The Technology Horizons report identifies: "the benefits of agility … allow many systems to swing from high-end, general-purpose applications to lower-end irregular warfare applications" (US Air Force 2011). This implies that the system can be modified to operate in different missions or "self-adapt as the environment in which the system is operating changes" (US Air Force 2011). This capability is addressed repeatedly, in reports from the Department of Defense (DoD) (Department of Defense 2012), the US Air Force (US Air Force 2011), and the Air Force Research Laboratory (AFRL) (Air Force Research Laboratory 2013) using different terms related to the same goal: agility.

The DoD also uses the terms 'resilience' and 'adaptive' for other desired features. The terms are combined in the DoD's description of directability: "directability (both to specify objectives but also how to adapt to the unexpected)" (Department of Defense 2012). Again, neither of these terms is quite sufficient as an alternative for the term 'agility'.

Other authors discuss agility by inverting the concept. They call this 'brittleness' (Department of Defense 2012), (Clare et al. 2012), (DePass et al. 2011), (Klein et al. 2004), (Christoffersen and Woods 2002), (Woods and Hollnagel 2006). "Inevitably, robot capabilities will exhibit brittleness as situations develop beyond their boundary conditions … these represent challenges to the adaptive power or resilience …" of the robot and human-robot team (Woods and Hollnagel 2006). If a design is brittle, it results in unnecessary performance tradeoffs and "additional manpower, vulnerabilities and lack of adaptability for new missions" (Department of Defense 2012). Although there were numerous references to 'brittleness' in the literature, we frame this feature more positively by proposing the term 'agility'.

The delineation between the related characteristics (robustness, flexibility, resiliency, and adaptability) and agility is that an agent may possess any or all of these characteristics, but lack agility. However, if the agent is agile, it does possess at least one of those characteristics. The difference is the corresponding timeframe. All of the characteristics contribute to agility, but unless that characteristic can be engaged quickly, the agent is not agile. The timeframe that drives agility is subjective, and depends on the operator, the domain/environment, and the system itself. The operator and/or agent developers define the degree of agility using this subjective context, as there is no globally attributable timeframe for agility. If the agent's response is considered quick enough, meeting the domain-specific time requirements, then the agent is perceived as agile.

Our definition of agility seeks to separate the terms agility, robustness, resiliency, adaptability, and flexibility. They are not all the same thing, though they are related. For our taxonomy, agility is the ability of the agent to leverage one or more of these characteristics to adjust the agent's behavior, and do so quickly.

Directability

Directability is a term used when describing autonomous agents. Various authors/literature describe directability as:

> The ability, "in which a human supervisor can define policies to influence agent activities at execution time" (Morley and Myers 2001).

The "means for controlling aspects of agent autonomy in a fashion that can be both dynamically specified and easily understood" (Klein et al. 2004).

"… giving the users the ability to substantively influence the machine agent's activities" (Christoffersen and Woods 2002).

Being achieved "by providing multiple mechanisms by which users can modify default assumptions and guide problem solution" (Truxler et al. 2012).

All of the above descriptions imply an external control to the agent that influences its behavior. Inspired by these, our definition of directability is:

Directability–*the ability for some external party to influence a change in the operation of an agent in order to accomplish a specific output or objective.*

Our definition matches the first half of the definition of directability in a DoD Defense Science Board Task Force Report: "both to specify objectives but also how to adapt to the unexpected" (Department of Defense 2012). The second half (i.e., how to adapt to the unexpected) of the report's definition is more aligned with our definition of agility.

For an agent to be directable, the first step is to identify which aspects of its functionality need to be directable. This, in turn, influences the agent's implementation. Directability can be at odds with autonomy and may require adjustment based on the desired level of autonomy. However, directability is a requirement in many system applications, so it is important to know what is directable in an agent in order to provide human operators with appropriate interfaces.

Observability

Many authors define, either explicitly or implicitly, the terms transparency and observability in such a way that they can be used interchangeably. We believe there is a distinct difference between the two and therefore will elaborate on the differences.

Rudolf Kalman defines observability in very mathematical terms and provides a mathematical proof for his definition of a plant's observability (Kalman 1959): "The state of a system is said to be 'observable' if the exact state of the system can be determined from measurements of the output signals over a finite time." However, we have simplified his definition slightly for our use:

Observability–*the level to which the exact state of the system can be determined from measurements of the output over a finite time.*

Observability enables "users to enter information, review default assumptions, and inspect and compare alternative COAs" (DePass et al. 2011). In other words, observability is the ability for an observer to measure the outputs of the system and understand what the agent has done, is doing, or will do soon.

Transparency

Authors typically do not define the word transparency when they use it, but imply its meaning in the context of the scenario they are describing or what transparency provides. "Machine transparency enables the human to understand what the machine is doing and [WHY]" (Air Force Research Laboratory 2013). For the human operator to understand what the agent is doing and why, agents "must be able to make pertinent aspects of their status and intentions obvious to their teammates" (Klein et al. 2004). In addition to the "why" of an agent's actions, some authors describe transparency as the insight or understanding of HOW the agent generated a solution, plan, action, etc. (Clare et al. 2012). Therefore, we use the following as the definition:

Transparency–*the ability to provide information about why and/or how an agent is in its current state.*

In more general terms, transparency is the degree to which an operator can understand what an agent is thinking. The Defense Science Board's Task Force Report on the Role of Autonomy in DoD Systems implies this meaning of transparency in its recommendation for the future development of autonomy: "The objective should be to create a technology base of diverse, platform-independent, transparent cognitive functions and tactics for integration into new missions" (Department of Defense 2012).

Some authors have either described transparency as observability or observability as transparency. In (Christoffersen and Woods 2002), observability is defined as "opening up the black box". Additionally, (Bass, Baumgart, and Shepley 2013) implied that observability gives the user insight into "the automation's judgment strategy" and, in turn, "found that human judgment performance improved when meta-information was provided regarding how the automation integrated input data to derive its judgment … compared to when only the judgment was provided." These descriptions match more closely to our definition of transparency, as opposed to observability, since it is giving the human more insight into why or how the agent is operating, not what its state happens to be at a given time. In contrast, it can be viewed that (Olson and Wuennenberg 2001) merge our definitions of observability and transparency when they describe visibility on a user interface.

Having more insight into why and how a system is operating helps the human operator anticipate agent behavior and calibrate one's trust in the agent's functionality. "For humans and machines to function as an effective team,

there must be an understanding of, and confidence in, its behaviors and decision making across a range of conditions" (Air Force Research Laboratory 2013). "If the robot is transparent to the user, than only one mental model is required … Thus, transparency is a desired element of efficient interaction" (Goodrich and Olsen 2003). "This understanding will enable an appropriate level of reliance on, or trust in, each team member for a given situation" (Air Force Research Laboratory 2013). When a user has trust in the system, then they will have more confidence when making decisions and directing the agent to perform its autonomous missions.

In sum, for these two descriptors, observability addresses *what* an agent is doing, and transparency is related to *how* or *why* an agent chose to do something.

Taxonomic Levels

Each of the descriptors defined above has five levels for classifying an agent (*None*, *Minimal*, *Limited*, *Moderate*, and *Complete/High*). The following are our initial thoughts on how these levels can be applied in describing the extent to which an agent possesses each descriptor. It is important to note, however, that these levels are influenced by the particular application domain. An agent may be defined as highly agile in one domain, but not agile in another. In other words, an agent designed for a certain domain may not retain the same descriptor level when applied to a different domain. This also means that any effort to improve an agent to achieve a higher level of a particular descriptor needs to take into account the application domain and how a change in the agent's computational techniques will, in turn, influence the level of that descriptor. Nevertheless, the domain does not need to be strictly scoped: agent developers and HMI designers do not need to agree on how to define the domain in order to use these descriptor levels. They do, however, need to agree on the salient features of the domain used in relation to this taxonomy.

Another factor to consider when applying these taxonomic levels is that a human's perspective also plays a role for interpreting agent descriptors. For instance, one person may view an agent 'completely transparent' if it is known that an agent will avoid obstacles that are in its path. A different person may want more detailed information for "complete transparency" (e.g., knowledge of the agent's algorithmic approach to obstacle avoidance). Another example pertains to agent observability: an agent developer may inform an HMI designer that complete observability cannot be provided for an obstacle avoidance technique because the vector fields cannot be displayed; an HMI designer may not need such detailed information to consider the agent observable. Despite these potential differences in interpretation, the proposed taxonomy and the accompanying descriptors and levels should improve communication between agent developers and HMI designers. The following details the proposed levels for each descriptor.

Agility Levels

Recall that our definition of agility is the ability of an agent to respond in an effective manner to new inputs within a short timeframe. The inputs could be realized as environmental changes, new or changed data (such as from sensors), or even changes within the agent itself. The levels of agility assigned to an agent are:

- *None*: if an agent cannot respond to changes. Also assigned if the agent cannot adapt to a change without restarting, as if from a clean slate. That is, it cannot leverage the existing state nor its existing knowledge to handle a change.
- *Minimal*: if the agent can be changed from its general implementation/configuration to handle very specific changes, but otherwise performs as if its agility rating is *None*.
- *Limited*: if the agent's current implementation can respond to changes for very specific cases.
- *Moderate*: if there are situations where the agent can generally respond to changes, but there are also situations where the agent cannot respond to those same changes.
- *High*: if the agent, under most situations, can consistently handle changes.

Directability Levels

Directability is the ability for some external party to influence a change in the operation of an agent in order to accomplish a specific output or objective. The emphasis on directability is the imposition of will: directability is related to the controlling agent or operator's *desired* output. The levels of directability assigned to an agent are:

- *None*: if the agent provides no ability for the human operator to achieve a targeted output via changing an agent's inputs ("inputs" include changes to variables/techniques in the agent's computational process).
- *Minimal*: if the operator can cause changes to some agent outputs, decisions, or actions via changing inputs, but in a way that may not reflect the operator's desired output.
- *Limited*: if the operator can manipulate a subset of the agent's inputs to change a subset of the agent's outputs to reflect the operator's desired output.
- *Moderate*: if the operator can make the agent achieve desired outputs via manipulation of the agent's inputs, but only indirectly.
- *Complete*: if the operator can make the agent achieve desired outputs via manipulation of the agent's inputs directly.

Observability Levels

Observability is the level to which the exact state of the system can be determined from measurements of the output over a finite time. Observability and transparency (described next) are tightly coupled. For instance, human operators typically want to increase their knowledge about the system and want to know more about both its state (observability) and *why* it is in that state (transparency). For the descriptor of agent observability, the proposed levels are:

- *None*: if the agent is a "black box", in which it only shows the outputs and nothing that helps the operator understand what the agent is doing, has done or is going to do.
- *Minimal*: if the agent can show the operator some of what it is doing, but additional state understanding cannot be obtained, even with multiple observations.
- *Limited*: if the agent can show the operator what it is doing, and support requests to obtain more state understanding, with a number of observations.
- *Moderate*: if the agent can provide enough information on what it is doing, such that, over a series of observations, its state can be completely understood by an operator.
- *Complete*: if the agent can show its complete state to an operator. Note: this does not mean that the internal state is shown, although it is possible. Ideally, to meet the 'complete' level, this requirement needs to be communicated before the agent development process begins. Having complete observability typically results in increasing the agent's level of transparency as well (see next).

Transparency Levels

Transparency is the ability to provide information about why and/or how an agent is in its current state. Typically, an operator's understanding of the underlying mechanisms within the agent and its current state is what increases transparency. Increases in transparency can come from operator knowledge and increased observability of the agent's state through time. Therefore, observability and transparency are coupled. Additionally, one can view transparency as a measure of the likelihood of the agent acting as expected. The levels for an agent are:

- *None*: in cases where the agent is given inputs and generates an output with no straightforward indication on how or why the agent produced the output.
- *Minimal*: if there is no observability into the state of the agent and the limited understanding of the agent's behavior extends from the operator's general knowledge of the underlying system.
- *Limited*: if observation and knowledge of the underlying mechanisms is required to determine why and how the agent is in its current state, but there are some elements that may produce unexpected results.
- *Moderate*: if observation and knowledge of the underlying mechanisms is required to determine why and how the agent is in its current state.
- *Complete*: if the internal computations can be made available for a user to understand how and why the system behaves in the way that it does.

Discussion

Agent developers and HMI designers can leverage the descriptors and their levels to improve the design of the joint agent/HMI system. To illustrate this concept, we provide a few simple examples.

Example 1: Route Planner Agent

In a tri-service autonomy focused research initiative that AFRL leads, we have an autonomous ground vehicle agent that can generate a route from its current location to a target location using a road network. The HMI for this supervisory control application provides the operator a "god's eye" view of the domain on a map (Figure 1). The agent development team knows the algorithms and the basic requirements of the domain (e.g., navigate to the target location), and the HMI team knows the agent can perform this navigation. With this most basic knowledge, the HMI is limited to the selection of a location to which the agent navigates.

Figure 1: Sample ground route shown on a map in simulation.

The agility of the system is based heavily on the algorithm(s) in use. In this example, the agent developer chose Dijkstra's algorithm and built an all-to-all mapping of the domain, performed when the agent starts. Then, all routes are a simple lookup from the existing table. This makes the agent response fast, but it severely affects agility. It prevents the agent from responding to extenuating events, such as blocked or slow roads. Thus, the agility level that this approach provides is *None*: for each new input (e.g., blocked road), it has to restart Dijkstra's algorithm to solve its route. Additionally, the level of directability it provides is *None*: the agent does not deviate from its shortest route to a target location. However, it can have *Moderate* observability, since the distance metric in use is simple to

understand. Finally, transparency is *Complete*, since the decisions regarding the route selection are easy to understand.

Now, consider the case where HMI designers wish to improve the agent's agility and directability: they want the agent to handle blocked roads and to visit certain locations enroute to its target location. The agent then must be reworked to comply with these new requirements. This rework is better scoped since it is clear what the HMI designers intend: increase both agility and directability to the *Limited* or *Moderate* levels, and keep the rest of the levels the same as much as possible. In response, the agent developer changes the routing algorithm to A*. This takes more time to calculate for each request, but it is still sufficiently fast to provide adequate agility for the domain. Given its speed, and its on-demand activity, it increases agility to *Limited*: it can perform a calculation based on a blocked road, as it is removed from the agent's list of candidate edges. It also increases Directability, since the operator could exclude roads to visit. The improvement in Directability that the HMI designer desired was to route the vehicle through certain points. The agent developer then provides a way to chain the target locations, where the agent solves for a route to the first location, then from the first to the second, etc. This helps to realize *Complete* Directability in defining the points to visit, along with *Limited* Directability in avoiding certain roads. This additional *Limited* Directability was not requested by the HMI designers, but has emerged as an additional feature that the agent can provide. The other taxonomic levels remain the same. Thus, with a change in the algorithm and some sacrifice in the runtime, the agent shifted from a precomputed solution to an on-demand solution. This increased agility and directability in a way that was desired by the HMI designers, plus provided an additional feature that the HMI designers did not anticipate. Given this taxonomy's common language for describing these features in the context of the domain, these improvements were facilitated and attention can then shift to designing interfaces by which the operator communicates more detailed route requests (e.g., which roads are blocked) to the agent.

The above example illustrates how our taxonomy can be applied such that a simple agent in a well-understood domain can be improved to support HMI requirements. It also describes how feedback between the HMI and agent developer communities helped to define the desired features of the agent in a clear, concise manner, which assists in identifying needed developments. While more work is involved in implementing changes to the agent and HMI, the overall capabilities of the resulting system for future, complex-task environments will be enhanced by the improved coordination between the operator and autonomous agents.

Example 2: Sense and Avoid Agent

This example pertains to a sense-and-avoid (SAA) agent developed for AFRL that is designed to provide autonomous conflict avoidance capabilities for unmanned vehicles by monitoring for potential conflicts with other aircraft in the airspace and providing avoidance steering. The SAA agent's agility is *High* as it is highly reactive to new information obtained from sensors. It is rated *Limited* for Observability, since the assembled trajectories can be shown. However, these trajectories are subject to hysteresis that adversely affects the observability. Since the resulting trajectory and the sensor input are fairly straightforward to correlate, its Transparency is *Moderate*. Its Directability, however, is *Minimal*, since the operator has little control over how the system responds to inputs.

AFRL has determined that increases in the agent's Directability is needed, such that the operator can improve SAA in response to changes in the operational environment. Specifically, it is desirable for the operator, via the HMI, to specify avoidance rule priorities and values (e.g., change well-clear distance), adjust the minimum separation threshold for maneuvers, and select which avoidance rules can be violated (e.g., right of way).

However, given that the SAA agent was initially developed to be autonomous, independent of operator input, a large part of its core functionality was built under the assumption that no HMI would be needed, and if one was needed, that the HMI would not need to interface with the agent's processing. If Directability had been considered early enough, the mechanisms leveraged by the SAA algorithm could have been made adjustable. The resulting system is very limited and any changes to increase Directability now (e.g., to provide access to parameters that are to be operator-adjustable) will be very difficult and time consuming.

This is an example where we feel the use of our taxonomy and descriptors early in the design process to support communications between agent developers and HMI designers would have resulted in a SAA agent with more Directability. Additionally, our taxonomy can be used to describe the limitations of the SAA agent in regards to an HMI, which would help in managing any expectations of possible HMI implementations.

Conclusion

Taxonomies for human-robot or human-machine interaction exist, but most are compositional in nature. They serve to improve understanding of the relationship between operators and systems, but provide little guidance for developing HMI that supports effective coordination between the operator and autonomous agents. Our proposed taxonomy provides a common language to support dialogue be-

tween agent developers and HMI designers to identify needed system functionalities earlier or describe current system functionalities and limitations. This taxonomy is by no means comprehensive, nor is it truly objective in nature: user bias, knowledge, and experience may influence the user's subjective interpretation of taxonomic levels. Also, all aspects of the taxonomy are tempered against the application domain, which should be well understood.

Our descriptors may also be useful for efforts focused on verification and validation (V&V) of autonomy. Often, autonomous agents are built to emphasize Agility and Directability without heed to the Transparency and Observability needed to build the operator's trust in the system. Those that focus on maximizing Transparency and Observability (to maximize the potential for V&V) often sacrifice Agility and Directability. Ideally, given some instantiated agent, there is a middle ground where Observability and Transparency can be improved while also maximizing the Agility and Directability.

Future work includes leveraging this taxonomy within our projects and determining what revisions are needed. In particular, the descriptors will be employed in dialog between our agent developers and HMI designers on projects that aim to provide HMI that support more effective coordination between the operator and autonomous agents.

Acknowledgements

This work was funded by the Air Force Research Laboratory (711 HPW/RHCI) under Contract FA8650-14-D-6500, Task 002.

References

Air Force Research Laboratory. 2013. Air Force Research Laboratory Autonomy Science and Technology Strategy. 88ABW-2013-5023.

Bartram, L. and Ovans, R. 1995. A dialogue-based approach to the design of user interfaces for supervisory control systems. In International Conference on Systems, Man and Cybernetics, 3144-3149.IEEE.

Bass, E. J.; Baumgart, L. A.; and Shepley, K. K. 2013. The Effect of Information Analysis Automation Display Content on Human Judgment Performance in Noisy Environments. *Journal of Cognitive Engineering and Decision Making.* 49-65.

Burke, J. L.; Murphy, R. R.; Rogers, E.; Lumelsky, V. J.; and Scholtz, J. 2004. Final Report for the DARPA/NSF Interdisciplinary Study on Human-Robot Interaction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 103-112.

Christoffersen, K. and Woods, D. D. 2002. How to make Automated Systems Team Players. *Advances in Human Performance and Cognitive Engineering Research.* 1-12.

Clare, A. S.; Cummings, M. L.; How, J. P.; Whitten, A. K.; and Toupet, O. 2012. Operator Objective Function Guidance for a Real-Time Unmanned Vehicle Scheduling Algorithm. *Journal of Aerospace Computing, Information, and Communication.* 161-173.

Department of Defense. 2012. Defense Science Board Task Force Report: The Role of Autonomy in DoD Systems.

DePass, B.; Roth, E.; Scott, R.; Wampler, J.; Truxler, R.; and Guin, C. 2011. Designing for collaborative automation: A course of action exploration tool for transportation planning. In Proceedings of the 10th International Conference on Naturalistic Decision Making (NDM 2011).

Dudek, G.; Jenkin, M. M.; Milios, E.; and Wilkes, D. 1996. A Taxonomy for Multi-Agent Robotics. *Autonomous Robots.* 375-397.

Farinelli, A.; Iocchi, L.; and Nardi, D. 2004. Multirobot Systems: A Classification Focused on Coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.* 2015-2028.

Gerkey, B. P. and Mataric, M. J. 2004. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *International Journal of Robotics Research.* 939-954.

Goodrich, M. A. and Olsen, D. R. 2003. Seven principles of efficient human robot interaction. In IEEE International Conference on Systems, Man and Cybernetics, 3942-3948.

Humphrey, C. M.; Gordon, S. M.; and Adams, J. A. 2006. Visualization of multiple robots during team activities. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 651-655.SAGE Publications.

Kalman, R. 1959. On the General Theory of Control Systems. *IRE Transactions on Automatic Control.* 110-110.

Klein, G.; Woods, D. D.; Bradshaw, J. M.; Hoffman, R. R.; and Feltovich, P. J. 2004. Ten Challenges for Making Automation a "Team Player" in Joint Human-Agent Activity. *IEEE Intelligent Systems.* 91-95.

Korsah, G. A.; Stentz, A.; and Dias, M. B. 2013. A Comprehensive Taxonomy for Multi-Robot Task Allocation. *The International Journal of Robotics Research.* 1495-1512.

Morley, K. L. and Myers, D. N. 2001. Human directability of agents. In Proceedings of the International Conference on Knowledge Capture.

Olson, W. A. and Wuennenberg, M. G. 2001. Autonomy based human-vehicle interface standards for remotely operated aircraft. In Digital Avionics Systems, 2001. DASC. 20th Conference, 7D3/1-7D3/9 vol.2.

Scholtz, J. 2003. Theory and evaluation of human robot interactions. In Proceedings of the 36th Annual International Conference on System Sciences, 10.Hawaii:.

Seneler, C. O.; Basoglu, N.; and Daim, T. U. 2008. A taxonomy for technology adoption: A human computer interaction perspective. In Portland International Conference on Management of Engineering & Technology, 2208-2219.

Steinfeld, A.; Fong, T.; Kaber, D.; Lewis, M.; Scholtz, J.; Schultz, A.; and Goodrich, M. 2006. Common metrics for human-robot interaction. In Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction, 33-40.

Steinfeld, A. 2004. Interface lessons for fully and semi-autonomous mobile robots. In Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, 2752-2757 Vol.3.

Truxler, R.; Roth, E.; Scott, R.; Smith, S.; and Wampler, J. 2012. Designing collaborative automated planners for agile adaptation to dynamic change. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 223-227.SAGE Publications.

US Air Force. 2011. Technology Horizons: A Vision for Air Force Science & Technology during 2010-2030 Volume 1. AF/ST-TR-10-01-PR.

Van Dyke Parunak, H.; Brueckner, S.; Fleischer, M.; and Odell, J. 2004. A Design Taxonomy of Multi-Agent Interactions. 123-137.

Woods, D. D. and Hollnagel, E. 2006. *Joint Cognitive Systems: Patterns in Cognitive Systems Engineering* CRC Press.

Yanco, H. A. and Drury, J. L. 2002. A taxonomy for human-robot interaction. In Proceedings of the AAAI Fall Symposium on Human-Robot Interaction, 111-119.

Yanco, H. A. and Drury, J. L. 2004. Classifying human-robot interaction: An updated taxonomy. In IEEE International Conference on Systems, Man and Cybernetics, 2841-2846.