Adaptive Autonomy for a Human-Robot Architecture

Stéphane Mercier, Catherine Tessier and Frédéric Dehais

ONERA & ISAE
2 - 10 avenue Edouard Belin
31055 Toulouse Cedex 4

Abstract

In the context of supervisory control of one or several artificial agents by a human operator, the definition of the autonomy of an agent remains a major challenge. When the mission is critical within a dynamic environment, e.g. in the case of uninhabited vehicles, errors are not permitted while performance must be as high as possible. Therefore a trade-off must be found between manual control, usually ensuring good confidence in the system but putting a high workload on the operator, and full autonomy of the agents, often leading to less reliability in uncertain environments and lower performance. Having an operator in the decision loop does not always grant maximal performance and safety anyway, as human beings are fallible. Additionally when an agent and a human decide and act simultaneously using the same resources, conflicts are likely to occur and coordination between these heterogenous entities is mandatory. We present the basic concepts of an approach that will aim at dynamically adjusting the autonomy of an agent in a mission relatively to its operator, based on a formal modelling of mission ingredients.

Keywords

Adaptive autonomy, Authority sharing, Multiagent systems, Human-Machine interactions.

1 Context of the Study, Assumptions and Objectives

In this paper we focus on the autonomy of artificial agents (e.g. uninhabited vehicles, autopilots...) supervised by a human operator and achieving objectives for a given mission. Such agents evolve in a dynamic environment and face unexpected events. Consequently real-time reactions to these events in order to avoid dangerous situations and the loss of the agents themselves are compulsory. Additionally we consider systems where most of operational tasks can be associated with procedures, i.e. tasks must be executed in a precise order and respect strict constraints (as it is the case in aeronautics).

In an ideal context the agents would be able to achieve the mission completely independently from the operator, a case that is hardly likely to occur in reality. However this is a necessary ability for the agents as communication breakdowns between the agents and the operator may occur during the mission. Beyond this extreme case the agents may request the operator's help anytime for any task when an issue arises. On the other hand the operator her/himself must be free to intervene at any stage of the mission in order to adjust the agents'behaviours according to her/his preferences but also to correct their possible mistakes or improve their performance.

One of the main challenges is conflicts. The human operator's inputs may interfere with the agents'plans and break their consistency anytime, even if the inputs are intended to improve a task or to correct an agent's mistake. As an agent and the operator may both execute actions on their own, it is of great importance that they should remain coordinated so that they should not use the same resources at the same time for different purposes. For example if the autopilot of a UAV [1] and the operator simultaneously "decide" to move the vehicle in different directions, inconsistencies are very likely to appear in the flight and lead to an accident. Therefore conflicts must be detected and solved as soon as possible.

Finally our main objective can be summarized in the following question: why, when and how should an agent take the initiative? When the environment has changed and the agents' plan needs to be updated? When the operator's inputs are inconsistent with the procedures and with security constraints? Or when they create conflicts with the current goals?

2 STATE OF THE ART

While there is no universal definition of autonomy, this concept can be seen as a relational notion between entities about an object [5, 2]: for instance, a subject X is autonomous with respect to the entity Z about the goal G. In a social context entities like other agents or institutions may influence a given agent thus affecting its decision-making freedom and its behaviour [4].

In the context of a physical agent evolving in the real world (i.e. an uninhabited vehicle) under the control of a human operator, autonomy can be seen as the ability of the agent to minimize the need of human supervision and to act alone [20]: the primary focus is then rather the operational aspect of the autonomy than the social one. In this situation pure autonomy is just a particular case of the agent - operator relationship, precisely consisting in not using this relationship.

However in practice, as automation within complex missions is not perfectly reliable and is usually not designed to reach the defined objectives alone, human supervision is still needed. Moreover it seems that human intervention significantly improves performance over time compared to a neglected agent [11, 10] (see Fig. 1).

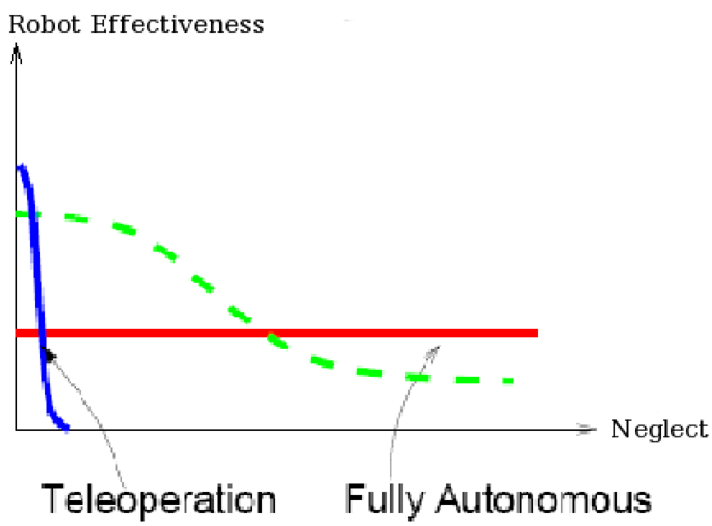

Figure 1: performance vs neglect

[1] Uninhabited Air Vehicle

Autonomy levels

[22] first proposed a classification for operational autonomy based on a ten-level scale. This model remains quite abstract as it takes into account neither the environment complexity nor the mission context. However it provides an interesting insight into the interactions between an operator and an agent. This model has later been extended, using the same scale applied to a four-stage cognitive information processing model (perception, analysis, decision-making and action) [18]. Based on the same principles other scales for autonomy classification have also been proposed, e.g. [1].

Other approaches aim at evaluating an agent's autonomy in a given mission context, like MAP [12], ACL [6] or ALFUS [14]. The latter proposes to evaluate autonomy according to three aspects: mission complexity, environmental difficulty and human interface. However this methodology aggregates many heterogeneous metrics and the meaning of the result is hard to evaluate. Moreover a qualitative step is necessary especially to set weights on the different tasks composing a mission and evaluate their importance.

The idea that operational autonomy can be graduated leads to the concept of adjustable autonomy or shared authority. The main principle is that machine and human abilities are complementary and they are likely to provide better performance when joined efficiently than when used separately [15]. A physical agent is thus capable of evolving at several predefined autonomy levels and switches levels according to the context. A level is defined by the complexity of the commands [8] or the ability to perform tasks without the need of operator's interventions [11]. The major limitation we can see in these approaches is the *a priori* definition of the levels, the static distribution of tasks among entities at each level and the fact that the number of levels is necessarily limited. Interactions between the agent and the operator are thus restricted to a given set and are determined by autonomy levels, there is no possibility of fine dynamic task sharing.

To add more flexibility, [19] endow agents with learning capabilities based on Markov Decision Processes (MDP) allowing them to better manage the need for human intervention. Agents can define their own autonomy levels from the user's provided intentions. However this method does not seem to be directly applicable to critical systems as the behaviour of learning agents facing unexpected situations is hard to validate. Moreover the operator's interactions are restricted to the agent's needs.

The approach of [17] adds more human control on the agent. Levels are not defined in a static way but come from a norm: permissions and restrictions describing the agent's behaviours are set by the operator. In order to do so, the operator has to create a complete set of rules like ”In case of medical emergency, consult the operator to choose landing location”. The major issues associated with such an approach are the high number of rules to provide and the risk of conflict between rules. Anyway the autonomy of the agent is completely human-supervised and the agent has no possibility to adapt by itself.

Sliding autonomy[3] consists in determining whether a task should be executed by the agent alone or by the operator using manual control; there is no direct reference to autonomy levels. Roles are not shared at the mission level but are reconsidered for each action to realize. However it seems that the range of human-agent interactions is really restricted as each task is performed either ”completely autonomously” or ”completely through teleoperation”.

In contrast, collaborative control is an approach aiming at creating dialogs between the operator and the agent [9]: the agent sends requests to the human operator when problems occur so that she/he could provide the needed support. This is again a restriction of all possible interactions: only dialog is used whatever the circumstances. In practice almost all interactions are initiated by the agent's requests and the operator acts almost exclusively as a support, she/he has not much initiative.

[21] have studied two authority sharing modes on a simulated space assembly task, SISA (System-Initiative Sliding Autonomy) where only the agent can request the operator's support and MISA (Mixed-Initiative Sliding Autonomy), where the operator can also intervene anytime. The allocation between the agent and the operator is realized separately for each task according to statistics to determine which entity will be the most efficient, which does not seem sufficient for a critical mission where errors are not allowed. However sharing at the task level is an interesting idea as it provides the most adaptive solution to the mission.

As shown by the literature review it is often interesting to join human and machine abilities to carry out a mission and adjustable autonomy seems a good principle. However the fact that the human operator also is fallible is often neglected. While it seems reasonable that the operator should keep the control over the agent, in most of the studies the operator's inputs are not evaluated and accepted "as they are" by the agent. Moreover the simultaneous decisions and actions from an artificial agent and a human agent might create misunderstandings and lead to conflicts and dramatic situations [7].

3 EXPERIMENTAL ENVIRONMENT AND SCENARIO

In order to validate our approach for adaptive autonomy and shared authority with concrete applications, the framework for experimentations in real conditions with human operators interacting with "autonomous" vehicles is already being designed.

3.1 The Scenario

The scenario is the localization and assessment of a fire by a UGV[2] in a partially unknown area. The mission for the UGV and the operator consists in looking for starting fires around a factory or a facility and determining its properties (localization, size, dynamics) so that it could be quickly put out. The area is hardly accessible, dangerous and partially unknown (no precise and updated map available). Additionally, the scenario could be extended with the possibility for the UGV to carry an extinguisher. This would allow the UGV to directly put out a very starting fire or delay a fire evolution in a given area, e.g. close to sensitive items. As the extinguisher would be very small, its use would have to be carefully chosen. Figure 2 shows the scenario.

Several operational assumptions are made:

- The area where the UGV evolves is divided into two parts: the start area which is known (a map is available), the search area which is partially unknown;
- the known area includes obstacles to avoid, but there are localized on a map;
- the human operator has no direct visual contact with either the UGV nor the outdoor environment;
- there are sensitive items in the known area, which have to be protected against the fire threat coming from the partially unknown area;

[2]Uninhabited Ground Vehicle

- the fires may evolve, possibly blocking known paths or endangering the UGV;
- a fire evolution is determined by the objects that can burn;
- the access paths to the search area are limited and narrow, making the access to the zone difficult.

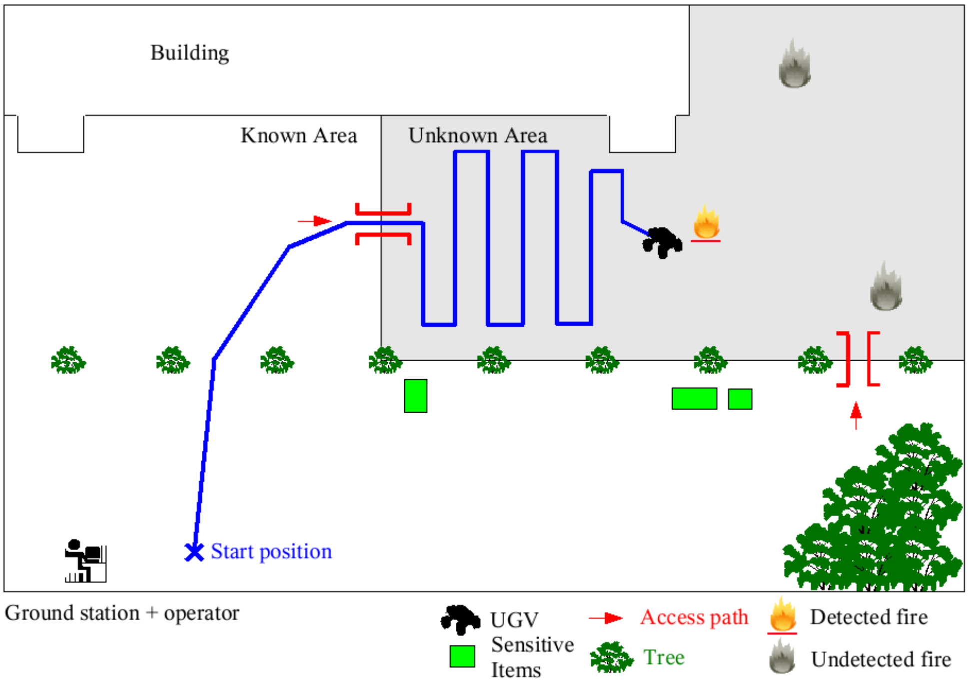

Figure 2: the scenario

Additionally, some hazards may impair the mission:

- communication breakdowns between the UGV and the operator;
- dynamic and uncertain environment in the search area (obstacles, fires);
- possible loss of GPS positioning;
- sensor failures.

3.2 The Experimental Set-up

ISAE[3] is developing an experimental set-up composed of a ground station and several Emaxx UGVs - see Fig. 3. The UGVs may be controlled either using a remote control (in case of problems) or a graphical interface (normal use). They carry several sensors (GPS, inertial sensors, scenic camera, ultrasounds, odometry) and are able to follow a set of waypoints autonomously. Algorithms are currently being developed to be implemented onboard (ARM 7 & 9 electronic cards, Linux OS) in order to give them decisional abilities (planning, situation assessment).

A wizard of Oz user interface is also being developed as it offers greater possibilities to control "unexpected" events during the experiments (e.g. : communication breakdowns, sensor failures).

[3]Institut Supérieur de l'Aéronautique et de l'Espace, resulting from the merging of the ENSICA and SUPAÉRO

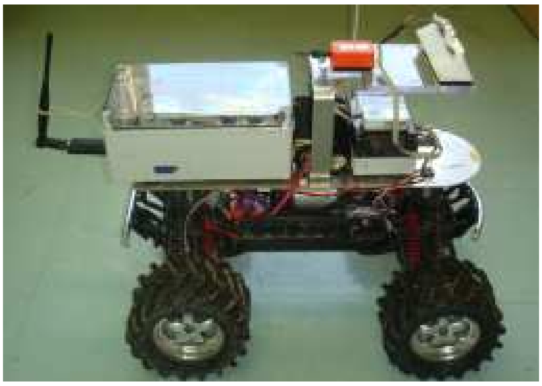

Figure 3: an Emaxx UGV

4 ARCHITECTURE FOR MISSION MANAGEMENT

Planning and situation assessment are the two key “high-level” functions for autonomous agents. However, a third function, the authority sharing function, is proposed so as to manage the interactions between an operator and an agent. This function will be based on conflict solving.

4.1 Planning

A mission consists in a set of *objectives* the agents should reach. To do so the agents will execute *tasks*, each task being supposed to provide an expected result while respecting some *constraints* (security, physical limits, authorizations, etc.) Each task that is executed uses and produces *resources*, and is a resource itself to reach an objective.

As the agent has to react to unplanned events occurring during the mission, the plan has to be continuously updated. This replanning process is a mandatory ability of the agent and is requested by the authority sharing function.

4.2 Situation Assessment

Situation assessment [16] constantly analyzes the state of the system: the current and possible future states are estimated according to the plan and procedures and to the evolution models of the environment, of the system itself and of all other relevant objects.

As far as the recognition of the operator’s intentions are concerned, the only available information comes from her/his inputs in the system. If a pattern is recognized from these inputs and can be associated with one or several known procedures, this constitutes a valuable clue about the operator’s non-explicit goals and may contribute to anticipate her/his future actions.

More precisely situation assessment compares the expected results of the tasks performed by the agents and the operator with the actual results and detects gaps that may appear. This allows potentials conflicts to be detected in the current state but also in an anticipated manner.

A conflict is a mismatch between the plan and its execution appearing as an inconsistency at the level resource. The situation assessment function identifies the conflicting resources and determines the characteristics of the conflict (involved entities, occuring time of the conflict, etc.)

Finally, this information is transmitted to the authority sharing function.

4.3 Authority Sharing

All issues in the execution of the mission correspond to a type of conflicts in the allocation of resources, between the entities or between the plan and its execution.

The authority sharing function gets information about conflicts from the situation assessment function: depending on the category of the conflict, appropriate solving methods can be executed. In any case, if a solution for a conflict exists, this is always realized through resource reallocation among the agents.

This can be done through the planning function. From the information coming from the situation assessment, the authority sharing function may add constraints to the planning process so that a new consistent allocation of resources can be found.

5 FIRST FORMALIZATION

The first step to get a formal and operational definition of adaptive autonomy is to formalize the basic concepts of a mission operated by physical agents and an operator.

Mission
A mission is a set of objectives to be reached by the agent(s) and the human operator.
Example:
M = { go to zone, detect fires, return to base }.

Resource
A resource is an item contributing to satisfying a mission objective. It can be a physical object, energy, a permission, a piece of information, a task, an algorithm, a logic condition...
A resource is written: $r = < item\_id, type, time\_interval, value[time\_interval], \mathcal{R}_{cons}, \mathcal{R}_{prod}, source >$,
with $item\_id$: the identifier of the resource;
$type$: the type of the resource (additive or absolute, exclusive, task, etc.);
$time\_interval = [t_{start}, t_{end}]$: the time interval defining the existence of the resource;
$value[time\_interval]$: a set of dated discrete values taken by the resource on $time\_interval$;
$\mathcal{R}_{cons}$: the set of resources consumed or needed by this resource;
$\mathcal{R}_{prod}$: the set of resources produced or affected by this resource;
and $source$: the origin of the resource (see definition below).

Example:
$r1 = < energy, additive, [10h05m17s - 10h06m20s], [1, 1, 1.5, ..., 1], \{battery\}, \{\}, source >$
with $source = < engine, 10h03m02s, Emaxx1 >$ (see definition of $source$ below).

Source
A source defines the origin of a resource.
A source is written:
$source = < r_{prod}, t_{prod}, a >$
with r_{prod} the producing resource;
t_{prod} the production time;
and a the producing entity (agent, operator...).

Example:
$source1 = < navigation1, 10h20m50s, Emaxx1 >$
$source2 = < piloting2, 11h42m20s, Emaxx1 >$

Tasks
Tasks are particular resources. They are created and scheduled by the planning algorithm in order to satisfy mission objectives.

Example:
Let $nav1$ be a task realized by the robot on operator's request. This resource is written:
$nav1 =< navigating, task, [t_{start} - t_{end}], initiated, \{map, navAlgorithm\}, \{waypointsList\}, src >$
with $\{map, navAlgorithm\}$ the resources needed to perform task $nav1$;
$waypointsList$ the resource produced by task $nav1$;
and $src =< GUIRequest, t_{prod1}, operator >$ specifying the origin of $nav1$, a request from the operator through the graphical user interface at t_{prod1}.
It is only when resources consumed by task $nav1$ have been allocated over time that this task takes the value $instantiated$ and its times t_{start} and t_{end} are set in the plan.
A *task* resource can take the following values:
$\{initiated, instanciated, executing, done, aborted, paused, failed\}$.

Conflicts
A conflict appears during the plan execution when inconsistencies at the resource level are detected. Situation assessment allows to detect such inconsistencies and to identify the involved resources. As all resources are marked with the field *source*, the producing entities are known. This makes it possible to classify conflicts in several categories, which are listed in table 1 (vertically the entity responsible for the plan disruption, horizontally the entity whose plan is affected).

Conflict	Operator	Agent	External World	Procedures
Operator	Contradictory simultaneous or successive Orders	Prioritary Order, contradictory to agent's plan	Physically unworkable Order	Violation (deliberately or not)
Agent	Prioritary Action, contradictory to operator's actions	Failure	Inconsistency between the agent's plan and actual measures	Violation (with operator's authorization or not)
External world	Invalidation of operator's action	Invalidation of agent's plan or actions	- *(external world is supposed to be consistent)*	Violation
Procedures	Modification of procedures being executed	Modification of procedures being executed	Inadapted Procedures	Procedure inconsistencies

Table 1: The different conflict categories

Several entities influence the mission : the human operator, the agent, the external world and procedures. The external world represents the world in which the physical agent evolves. Procedures represent the set of rules that the operator and the agent must abide by. They are established by external entities like designers of the system or regulation authorities (security rules for instance). The operator and the agent may violate these rules, deliberately or not.

6 CONCLUSION AND FUTURE WORK

We have presented the general principles and some basic concepts for an approach of operational adaptive autonomy. Resources, including the operator's tasks, are the key items to determine the contribution of each entity to the mission's objectives. Conflicts can be detected and classified depending on the entities that disrupt the plan. Consequently task reallocation within the system is performed so that conflicts could be solved safely with every entity being aware of what is being done.

Task reallocation will take into account the current capacities of the agents and operators, the operators'desires, the mission constraints, the priorities of objectives, all this being aggregated by the authority sharing function and transmitted to the planning function. Early conflict detection will allow agents to adapt their behaviours to the estimated operator's intentions as long as main constraints and objectives are respected, therefore improving the overall system performance. However, whether the operator intervenes or not, the agents are still expected to have the means to react "alone" to key issues.

Another aspect of adaptive autonomy is the fact that agents should be able to alleviate the operator's workload, e.g. relieving her/him of routine tasks and let her/him focus on key tasks of the mission. Again this is based on mutual situation monitoring and assessment and a better allocation of resources (including tasks) within the system when the context changes.

Current work focuses on a formal definition of mission execution including the dynamic aspects of the basic concepts we have defined, particularily conflicts, and on the fine identification of what precisely is involved in task reallocation. These concepts have to be operationalized then implemented on a real UGV platform (Emaxx) in order to conduct experiments with human operators[4]. Reliability, overall performance and the operator's satisfaction will allow us to assess our concepts for adaptive autonomy in real conditions.

References

[1] J.M. Bradshaw, M. Sierhuis, A. Acquisti, R. Feltovich, R. Hoffman, R. Jeffers, D. Prescott, N. Suri, A. Uszok, and R. Van Hoof. Adjustable autonomy and human-agent teamwork in practice: An interim report on space application. In *Agent Autonomy* [13], chapter 11.

[2] S. Brainov and H. Hexmoor. Quantifying relative autonomy in multiagent interaction. In *Agent Autonomy* [13], chapter 4.

[3] J. Brookshire, S. Singh, and R. Simmons. Preliminary results in sliding autonomy for coordinated teams. In *Proceedings of the AAAI 2004 Spring Symposium*, CA, 2004.

[4] C. Carabelea and O. Boissier. Coordinating agents in organizations using social commitments. *Electronic Notes in Theoretical Computer Science, Elsevier*, 2006.

[5] C. Castelfranchi and R. Falcone. From automaticity to autonomy: the frontier of artificial agents. In *Agent Autonomy* [13], chapter 6.

[6] B. T. Clough. Metrics, schmetrics! How the heck do you determine a UAV's autonomy anyway? In *Proc. of the Performance Metrics for Intelligent Systems Workshop*, MD, 2002.

[4]see section 3

[7] F. Dehais, A. Goudou, C. Lesire, and C. Tessier. Towards an anticipatory agent to help pilots. In *Proceedings of the AAAI 2005 Fall Symposium "From Reactive to Anticipatory Cognitive Embodied Systems"*, Arlington, Virginia, 2005.

[8] G. Dorais, P. Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems. In *IJCAI'99 Workshop on Adjustable Autonomy Systems*, Stockholm, Sweden, 1999.

[9] T.W. Fong, Ch. Thorpe, and Ch. Baur. Collaboration, dialogue and human-robot interaction. In *10th International Symposium on Robotics Research*, Lorne, Victoria, Australia, 2002.

[10] M. Goodrich, T. McLain, J. Crandall, J. Anderson, and J. Sun. Managing autonomy in robot teams: Observations from four experiments. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, 2007.

[11] M. Goodrich, D. Olsen, J. Crandall, and T. Palmer. Experiments in adjustable autonomy. In *Proceedings of Ijcai 2001 Workshop on Autonomy, Delegation and Control: interacting with autonomous agents*, 2001.

[12] B. Hasslacher and M. W. Tilden. Living machines. Los Alamos National Laboratory, 1995.

[13] H. Hexmoor, C. Castelfranchi, and R. Falcone. *Agent Autonomy*. Kluwer Academic Publishers, 2003.

[14] H. Huang, K. Pavek, B. Novak, J. Albus, and E. Messin. A framework for autonomy levels for unmanned systems (ALFUS). In *Proceedings of the AUVSI's Unmanned Systems North America 2005*, Baltimore, Maryland, June 2005.

[15] D. Kortenkamp, P. Bonasso, D. Ryan, and D. Schreckenghost. Traded control with autonomous robots as mixed initiative interaction. In *Proceedings of the AAAI 1997 Spring Symposium on Mixed Initiative Interaction*, March 1997.

[16] C. Lesire and C. Tessier. A hybrid model for situation monitoring and conflict prediction in human supervised "autonomous" systems. In *Proc. of the AAAI 2006 Spring Symposium "To Boldly Go Where No Human-Robot Team Has Gone Before"*, CA, 2006.

[17] K. Myers and D. Morley. Human directability of agents. In *K-CAP 2001, 1st International Conference on Knowledge Capture*, Victoria, Canada, 2001.

[18] R. Parasuraman, T. Sheridan, and C. Wickens. A model for types and levels of human interaction with automation. *Systems, Man and Cybernetics, Part A, IEEE Transactions*, 30(3):286–297, 2000.

[19] P. Scerri, D. Pynadath, and M. Tambe. Adjustable autonomy for the real world. In *Agent Autonomy* [13], chapter 10.

[20] D. Schreckenghost, D. Ryan, C. Thronesbery, P. Bonasso, and D. Poirot. Intelligent control of life support systems for space habitat. In *Proc. of the AAAI-IAAI Conference*, 1998.

[21] B. Sellner, F. Heger, L. Hiatt, R. Simmons, and S. Singh. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. In *Proceeding of the IEEE, 94(7)*, 2006.

[22] T.B. Sheridan and W.L. Verplank. Human and computer control of undersea teleoperators. Technical report, MIT Man-Machine Systems Laboratory, Cambridge, MA, 1978.