Authority management in human-robot systems

Stéphane Mercier [*] **Catherine Tessier** [*] **Frédéric Dehais** [**]

[*] *ONERA, 2 avenue Édouard Belin, 31055 Toulouse Cedex 4, FRANCE, (e-mail: surname.name@onera.fr).*
[**] *ISAE, 10 avenue Édouard Belin, 31055 Toulouse Cedex 4, FRANCE, (e-mail: frederic.dehais@isae.fr)*

Abstract: In the context of missions accomplished jointly by an artifical agent and a human agent, we focus on a controller of the authority dynamics based on a dependence graph of resources that can be controlled by both agents. The controller is designed to adapt the behaviours of the artificial agent or of the human agent in case of an authority conflict occurring on these resources. The relative authority of two agents regarding the control of a resource is defined so as the authority conflict, which appears relevant to trigger authority reallocation between agents as shown by a first experiment. Finally a second experiment shows that beyond the modification of the artificial agent's behaviour, it is also possible to adapt the human operator's behaviour in order to solve such a conflict.

Keywords: Adaptive autonomy, Authority sharing, human-robot interactions, Petri nets

1. INTRODUCTION

We consider an heterogeneous agent system in which a robot (UGV, UAV[1]) or a software agent (automatic pilot in a plane) accomplishes a mission interacting with a human agent (operator, pilot). During the mission, agents are given or take the authority Hardin and Goodrich (2009) to use a resource, to perform a task, to satisfy a goal: for instance, an aircraft automatic pilot in mode *Vertical Speed* has the authority on the flight control surfaces in order to reach a given altitude while guaranteeing some flight performance; if the crew disconnect the automatic pilot, then they take over the authority on the controls to reach this goal. A change in authority allocation can be planned in procedures or in the mission plan, or can be unexpected: this happens when the human operator takes over a task controlled by the artificial agent (software or robot) because she detects a failure, or for any reason of her own; or when the artificial agent takes over a task controlled by the operator because the operator's action violates some constraints (collision with an obstacle, etc.), or because communication with the operator is broken; or when no agent has the authority anymore: for instance, the automatic pilot has disconnected itself and the crew, who is not flying, is not aware of that.
The challenge created by these unexpected changes in authority allocation is that conflicts are likely to appear within the system, due to the fact that either the plan for both the human and artificial agents is not followed anymore, or the operator has a wrong situation awareness Wickens (2008), or both. This is illustrated by the following experiment, conducted at Supaero-ISAE, on a target search mission by a ground robot and a remote human operator. During the mission, the operator must pilot the robot manually to identify targets, *via* her interface. Whereas the operator takes over the robot for identification, the robot starts returning to base, a procedure triggered by the detection of a failing battery (unexpected event simulated by a wizard of Oz interface[2]). This event is presented on the operator's interface *via* three alarms: the battery icon switches from green to orange, the piloting mode blinks twice from "manual" to "supervised", and the display shows "Return to base" in green. However this unexpected event occurs at a critical time in the mission when the operator is particularly focused on the identification task in manual mode; therefore it is expected she will not perceive this change of states and that both agents (operator and robot) will persevere in pursuing their own goal (identify the target or return to base, respectively).

Table 1 shows the eye-tracker results of 14 subjects[3], analyzed with the Eye tech Lab software. The first column is the participant's identifier, the second one the conflict duration, the third one whether the reason of the conflict has been understood and the fourth one the seen alarms. For instance, participant Dasje took 6 seconds to detect and understand the conflict, he looked at least once at the "supervised" mode, the "return to base" display and the battery state. Participant Dupni did not understand the conflict after 50 seconds, although her gaze went on "supervised" and on "return to base". However she never looked at the state of the battery. The results of this experiment are consistent with our hypotheses as the authority conflict led 10 subjects out of 14 to persevere in their target identification task, which was an error. The analysis of the gaze behaviour of the 4 successful participants reveals that these operators looked at least at the couples ("battery state", "supervised" mode) or

[1] Unmanned Ground or Aerial Vehicle

[2] allowing to trigger events without the operator being aware
[3] subjects are equipped with a Pertech (25Hz) eye-tracker.

(”battery state”, ”return to base”) or even at the three pieces of information. Finally, it is worth mentionning that the robot sticked to its plan of returning to base, without taking into account the conflict with the operator.

Subject	Conflict duration (in s.)	Understanding	Seen alarms
Dasje	6	yes	supervised, return to base, battery
Deffra	50	no	supervised
Dupni	50	no	supervised, return to base
Gatthi	50	no	supervised, return to base
Guiju	50	no	-
Guiny	50	no	battery
Hosal	18	yes	return to base, battery
Jacchi	50	no	supervised
Nival	10	yes	supervised, battery
Peich	29	no	return to base
Penju	50	no	supervised
Pense	35	yes	supervised, battery
Rojan	50	no	-
Schpa	50	no	-

Table 1. Participants’behaviours during the authority conflict

The work presented here aims at designing and implementing an *authority dynamics controller* within the artificial agent architecture, in order to: 1/ detect conflicts related to an unexpected authority change; 2/ identify the consequences of such conflicts on the mission; 3/ adapt the agents’plans and if necessary, the authority allocation, in order to ”accept” the authority change or on the contrary to ”counter” it; the adaptation encompasses the sending of information or guidance to the operator.

We suppose that the artificial agent is equipped with a planning function (or a set of procedures) and a situation assessment function (or state estimation). The authority dynamics controller will be based on objective mission elements (software and physical resources, tasks, goals, constraints) which will be called *resources*. On top of each resource we will define the authority of an agent relatively to another agent. The state of these resources is an abstraction of the system state provided by the siuation assessment function. The mission plan (or procedure) is extracted as a *resource graph*, where conflicts caused by an unexpected authority change will appear. Conflict solving will consist in modifying the resource graph *via* a plan modification or an authority change between agents on a resource subset.

2. AUTONOMY AND AUTHORITY

The literature is mainly focused on predefined autonomy levels that are descriptive and not suited to real operations of robots. Sheridan and Verplank (1978) first proposed a classification for operational autonomy of a robot system based on a ten-level scale. This model remains quite abstract as it takes into account neither the environment complexity nor the mission context. Other scales for autonomy classification have been proposed, e.g. Bradshaw et al. (2003). Other approaches aim at evaluating the autonomy of a robot in a given mission context, like MAP Hasslacher and Tilden (1996) or ALFUS Huang et al. (2005). 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.
The main principle of these approaches is that machine and human abilities are complementary and they are likely to provide better performance when joined efficiently than when used separately Kortenkamp et al. (1997). A robot 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 Dorais et al. (1999) or the ability to perform tasks without the need of operator’s interventions Goodrich et al. (2001). The major limitations we can see in these approaches is the *a priori* definition of the levels and the static distribution of the tasks and interactions between the robot and the operator at each level.
To add more flexibility, Goodrich et al. (2007) distinguish between adjustable autonomy, where the operator chooses the operating modes of the robot, and adaptive autonomy, where the robot itself chooses its operating mode. Scerri et al. (2003) endow robot agents with learning capabilities allowing them to better manage the need for human intervention. 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 needs of the robot agents. On a similar principle, Schurr et al. (2009) build a model allowing artificial agents and human operators to transfer decision making to each other and compare their decisions. Inconsistencies in the team can be detected in order to be solved. While the idea of inconsistencies seems to be really relevant in the context of a team of agents, the authors do not say how they should be solved (who should have the priority if the artificial agent and the human operator disagree?)
In contrast, collaborative control is an approach aiming at creating dialogs between the operator and the robot Fong et al. (2002): the robot sends requests to the human operator when problems occur so that they 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 robot and the operator acts almost exclusively as a support. Sellner et al. (2006) base task allocation between the robot and the operator on statistics to determine which entity will be the most efficient. This does not guarantee a success because statistics summarize very different situations. However authority 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. Moreover the simultaneous decisions and actions of artificial and human agents are likely to create misunderstandings and lead to conflicts Dehais et al. (2005). Indeed, to our knowledge, control changes are not studied under the perspective of the conflicts that they can create between agents. As the experiment presented in introduction shows, as well as the study of aviation accident reports Dehais et al. (2005), unexpected or misunderstood authority changes can lead to inefficient, dangerous or catastrophic situations. In order to consider the human agent and the artificial agent in the same way, we prefer to use the concept of authority and authority control instead of autonomy, which concerns the artificial agent exclusively. The authority dynamics controller that we present aims at detecting and solving authority conflicts, taking into account the following different requirements about control: the initiative can be given to the artificial agent, to the operator, or to both agents; the granularity of authority objects must be appropriate, for detecting and solving conflicts. Moreover there is a need for an objective and application-independent criterion of authority evolution. Finally, models of the operator's tasks and "state" should be incorporated into the artificial agent's knowledge, as well as the operator's inputs (from "low level" orders to "high level" orders).

3. RESOURCES AND AUTHORITY

3.1 Resources

An authority conflict appears as an inconsistency in an agent's plan (for instance a constraint is not respected), or as an interference between several agents'plans (for instance, the use of the same non-shareable resource at the same time). In order to possibly reallocate authority between agents, the conflict must be evaluated: the involved agents must be identified, as well as the violated constraints or the part of the plan that is impaired. The authority dynamics controller will be based on a representation of the agents current plans, i.e. which tasks they perform, which means they need, which constraints they must respect to satisfy their goals. Therefore we represent an agent's plan as a dependency graph. We suppose that the artificial agent (that will embed the authority dynamics controller) has planning abilities. A dependency graph is obtained from the abstraction of a plan generated by an automated planner (e.g. an HTN planner - Hierarchical Task Network - like JSHOP2 Nau et al. (2001)). Dependencies are extracted from the planning operators, as they can be linked to each other by matching their instanciated preconditions and effects.

Definition: the dependency graph resulting from the abstraction of a plan is a graph whose nodes are *resources* and arcs represent the precondition relationship "A needs B" between resources. Therefore a resource represent a physical element of the system (sensor, energy, etc.) or a symbolic element (piece of information, task, goal, constraint, operator's input, etc). A leaf-node is a goal. Such a dependency graph will be referred to as *resource graph*. This is a dynamic graph that will be updated according to the updates of the plan during the mission. In our formal model based on Petri nets, resources are represented using a single generic net, encompassing a resource states and static properties. As dependency arcs can also have different properties, we represent them using another generic Petri net that we call *interface*, see Mercier et al. (2009).

Example: in order to produce the goal "Reach WP-Goal", the planning operator *NavigationRobot* is instanciated with the parameter *WP-Goal* as the destination and robot as the performing agent. We thus obtain the resource graph shown on figure 1, with *WP-Goal* as the goal, *Navigation Robot* as the instanciated task and the energy, position, steering wheel and safety distance as the preconditions. We notice that if one of the resources vanishes during the execution due to an unexpected event, the depending resources will be affected too.

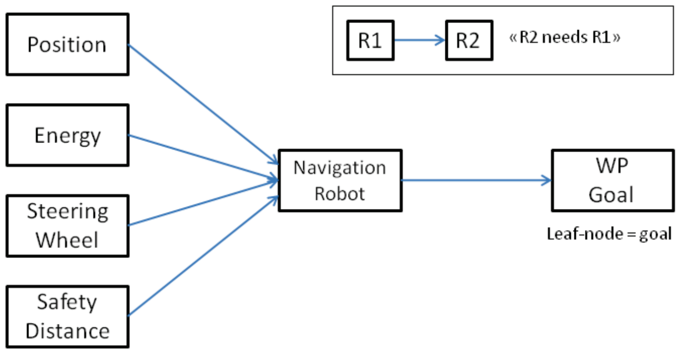

Fig. 1. Resource graph to reach goal *WP-Goal*

3.2 Authority

Definition: the authority of an agent X on a resource R relatively to another agent Y is defined through the following properties:

- the *access* right is the capacity of agent to X include a resource R in a resource graph, that is to control R in order to reach a goal;
- the *preemptability* right: for a "non shareable" resource, though X has access to R, it can be controlled by Y. The preemptability right gives agent X the right to use R as soon as needed, taking it from agent Y if necessary;
- the *control guarantee* right: once agent X controls R (i.e. X has accessed the resource), X may lose it to the benefit of Y through preemption. The control guarantee allows agent X to be certain that agent Y will not be able to take R away from it.

Therefore the authority of an agent on a resource is characterized by:

- a *gradation* of the agent's authority: agent X's control on resource R gets stronger as it is granted with the access right, the preemptability right and the control guarantee right, in this order.
- authority, as autonomy Castelfranchi and Falcone (2003) is a *relative* concept: for instance, for a given resource R, agent X may have the preemption right over agent Y, but not over agent Z. Consequently there are as many authority relationships as there are couples of agents that may control the resource.
- authority is *shared* between the agents: for a couple of agents $< X, Y >$ that may control resource R, the authority gain of agent X on resource R corresponds to an authority loss for agent Y. For instance, if agent X obtains the control guarantee on R, this means agent Y loses preemptability. Consequently if agent X gets exclusivity rights on R, agent Y will not have

access to R anymore : agent X *prevents* agent Y to access resource R, even if it does not use it.

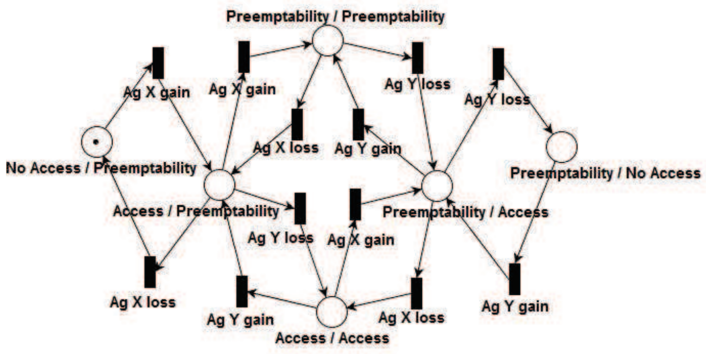

Fig. 2. Authority relationship between two agents X and Y on a given resource R: Petri net $\mathrm{A_R(X, Y)}$.

The Petri net $\mathrm{A_R(X, Y)}$ in figure 2 represents the authority relationship between two agents X et Y for a given resource R. Each place corresponds to the state of access, preemptability and control guarantee for both agents X and Y regarding resource R. The state changes modify the behaviour of R, i.e. it determines whether R can be allocated to X or Y, or not.

Agent	Authority	Access	Preemption	Guarantee
X	No Access	No	-	-
Y	Preemptability	Yes	Anytime	Yes
X	Access	Yes	Waiting	No
Y	Preemptability	Yes	Anytime	Yes
X	Preemptability	Yes	Anytime	No
Y	Preemptability	Yes	Anytime	No
X	Access	Yes	Waiting	Yes
Y	Access	Yes	Waiting	Yes

Table 2. Properties of relative authority between agents X and Y for resource R

In Table 2, each double line gives, for each agent X and Y, its authority on resource R and the status of the three associated properties. There are two intermediate states for which the authority of the agents is equivalent, (Access / Access) and (Preemptability / Preemptability). As far as the first one is concerned, the agents cannot take the resource from one another, each one must wait for the other one to relax the resource. This is a *cooperation* context. As far as the second one is concerned the agents can take the resource control from one another indefinitely, which makes the behaviour of the system unefficient or even dangerous. This is a *competition* context.

4. CONFLICTS ON RESOURCES

4.1 Definitions

During the mission execution, discrepancies may appear between the plan and the observed facts. They may come from hazards occurring in the environment, failures or unplanned actions of the agents. They appear in the resource graph, as it is an abstraction of the nominal plan: one or several resources within the graph are put into an inconsistent state. The inconsistency for a resource is defined as a non-desired marking in the resource Petri net model. A non-desired marking can appear on any resource of the model. As resources are all represented by the same Petri net, it is only necessary to study the reachable markings of this net, according to its inital marking (i.e. the resource properties). A reachable marking results either from a nominal event sequence, or from an unexpected event sequence, the latter being undesired.

Definition: a *conflict* on a resource R is a non-desired reachable marking of the Petri net representing R. There are two types of conflicts:

- There is a $\mathrm{Conflict_{Destruction}}$ when R is in state *Absent* while it is *Allocated*. It corresponds to a faulty hardware, a software error or a task failure, depending on what the resource represents.
- There is a $\mathrm{Conflict_{Preemption}}$ when R is a non-shareable resource and has two simultaneous users. It follows the action of an agent who is requiring R, which is already used by another agent.

Example: the robot performs its navigation task to reach *WP-arrival*. The operator unexpectedly decides to take manual control of the robot trajectory *via* her joystick for heading control. The human agent's action modifies the resource graph, as shown on figure 3. The request is associated with the fact that the operator wants to perform a manual navigation task, whose arrival point is only known by the operator. Resource *Steering Wheel* that was already allocated to resource *Navigation Robot* via interface *int3*, is preemptable and requested by resource *Navigation Operator via* another interface: resource *Steering Wheel* is allocated *via* interface *int8*. Resource *Steering Wheel* is then simultaneously allocated to resources *Navigation Robot* and *Navigation Operator*, which is inconsistent as *Steering Wheel* is a "non shareable" resource. This is a $\mathrm{Conflict_{Preemption}}$. The resource graph of figure 3 explicitly represents the state of knowledge of the robot agent after the human agent's action.

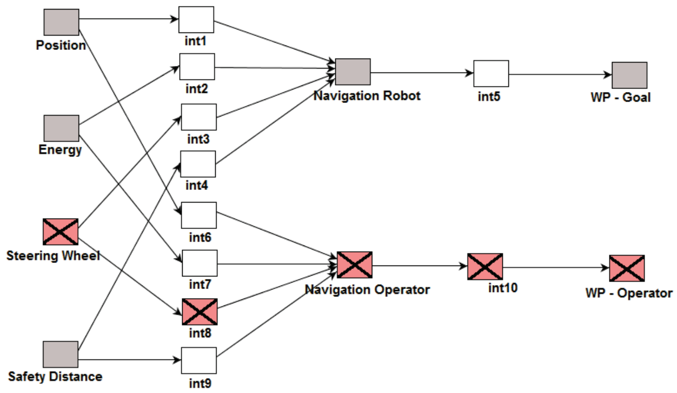

Fig. 3. resource graph after insertion of detected human operator's action, conflict

4.2 Authority conflict solving

Solving without authority change We have seen that there are two types of conflict: $\mathrm{Conflict_{Destruction}}$ and $\mathrm{Conflict_{Preemption}}$, which correspond to undesired reachable markings within one or several resources in the resource graph. The role of the authority dynamics controller, which is part of the robot agent, is to detect conflicts then to restore back the consistency within the resource graph:

- Conflict detection: an undesired marking is detected within R Petri net.
- Request to the planner: triggered by the authority dynamics controller. A new plan is searched to restore consistency by reallocating resources between agents, while keeping the existing goals. However some parts of the plan may be imposed, depending on the agents'actions that cannot be modified: the search space is constrained by the actions achievable by the agents as well as the authority relationships between agents on resources. The interaction with the human operator is one of the actions available to the artificial agent in order to restore the consistency of the resource(s) involved in the conflict.
- If a solving plan is found, this means that dependency arcs (interfaces) need to be deleted or created to update the resource graph. The existing authority relationships remain unchanged.

Example: on the example on figure 3, a preemption conflict has occurred on resource *Steering Wheel*. Resources *Navigation Robot* and *Navigation Operator* are competing for the control of resource *Steering Wheel* which is "non-shareable". $\mathrm{A}_{\mathrm{SteeringWheel}}(\mathrm{r},\mathrm{h})$ represents the authority relationship between the robot agent r and the human operator h on *SteeringWheel* (see figure 2). This authority relationship is such that the robot has Access to *Steering Wheel* and the human can Preempt *Steering Wheel*, so the interface created by the operator gets the priority over the interface created by the robot. However replanning must be triggered, as the robot agent's plan is now unfeasable: a plan satisfying both goals *WP-Operator* and *WP-Goal* is searched, the abstraction of which contains resource *Navigation Operator*.

Authority sharing dynamics If no solving plan is found without authority change, planning requires the modification of some authority relationships in order to release some constraints and increase the size of its search space: transitions *Ag r gain*, *Ag r loss*, *Ag h gain* and *Ag h loss* on relationships $\mathrm{A}_{\mathrm{R}}(\mathrm{r},\mathrm{h})$ can be fired, R being one of the conflicting resource or any other available resource allowing to create a solving plan. In case no solving plan is found within this enlarged search space, the planner must generate a downgraded plan, inducing the loss of one or several goals.

Example: While the human operator is controlling the manual navigation task (resource *Navigation Operator*), she violates the safety distance constraint, destroying resource *SafetyDistance*. There is a $\mathrm{Conflict}_{\mathrm{Destruction}}$ within resource *SafetyDistance*. A solving plan consists in urgently giving back the navigation control to the robot agent: transitions *Ag r gain* then *Ag h loss* are successively fired within $\mathrm{A}_{\mathrm{SteeringWheel}}(\mathrm{r},\mathrm{h})$, in order to give Preemptability to the robot: the robot agent has gained authority temporarily over the human operator for resource *SteeringWheel*. This makes it possible to allocate resource *Steering Wheel* to resource *EmergencyNavRobot* to produce resource *SafetyDistance* which became a goal.

5. EXPERIMENTS

The experiment described in the introduction is considered again (same experimental setup, same scenario), using the authority dynamics controller. This second experiment was conducted with 12 subjects. The goal was to test empirically one of the actions available to the authority dynamics controller to restore consistency in case of a conflict, using a planned interaction with the operator. Based on a simplified model of the operator's situation awareness (her gaze behaviour), the artificial agent attempts to solve the conflict by modifying the operator's behaviour instead of its own. As the human operator is persevering into her task, the robot agent uses an interaction procedure to change the information presented to the operator to get her out of her task. Figure 4 shows the resource graph corresponding to the robot returning to base while making the operator understand the robot's behaviour.
The artificial agent has two distinct simultaneous goals *WP-Base* and *SA_Operator*, which explains that the resource graph is not connected. The resource subgraph for goal *WP-Base* involves task *Navigation Robot* controlling *SteeringWheel*, which in turn implies that the robot has preempted *SteeringWheel* from the operator. The second goal *SA_Operator* means that the artificial agent aims at getting the human operator out of her perseveration by informing her about the conflict ($\mathrm{Conflict}_{\mathrm{Destruction}}$ on resource *Energy*) and about the plan generated to solve this conflict; the operator has to recover her situation awareness. Therefore this resource graph contains an interaction task *SendSACounterMeasure*, which is a cognitive counter-measure Dehais et al. (2003) based on the conflict information *Info:ConflictOnEnergy* and plan information *Info:NewPlanGoToWP-Base*. The eye-tracking data showed that these pieces of information had not been seen. The cognitive counter-measure consists in removing the panoramic vision display where the operator is focused, and to replace it during 4 seconds with the message "Battery failure, robot returning to base".

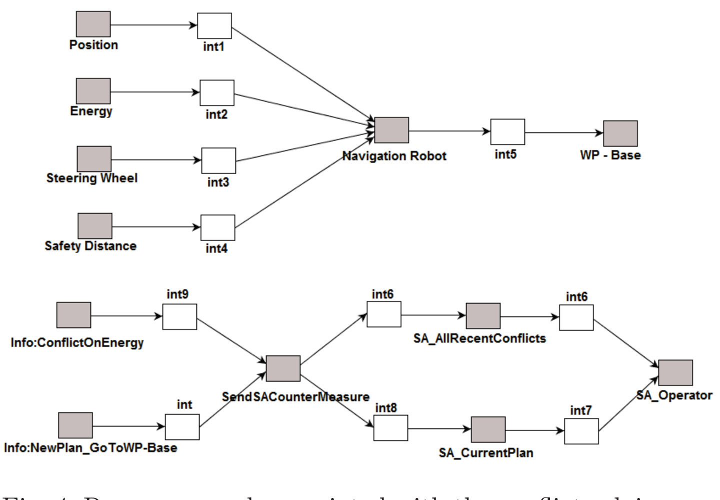

Fig. 4. Resource graph associated with the conflict solving

Table 3 summarizes the second experiment results. The first column indicates the subject id, the second one the conflict duration after sending the counter-measure and the third one details if all three alarms have been looked at least once since counter-measure sending. For example, participant Pouch immediately perceived and understood the conflict after counter-measure sending; and the counter-measure led him to look at least once at the three alarms. Participant Garu perceived all the relevant information but it took him 26 seconds to release complete control to the robot. The result analysis shows that this concept of interaction with the human operator, triggered by the authority dynamics controller, is efficient: 10 participants out of 12 gave immediately the control back to the robot to let it return to base automatically. The two remaining participants said that they had perceived the conflict with the counter-measure but thought they had time before the battery would be totally discharged.

Subjet	conflict duration (in s.)	Seen alarms
Pouch	0	supervised, return to base, battery
Gargu	26	supervised, return to base, battery
Lesc	0	supervised, return to base, battery
Berdo	0	supervised, return to base, battery
Thola	0	supervised, return to base, battery
Lasni	0	supervised, return to base, battery
Gabje	0	supervised, return to base, battery
Jacra	0	supervised, return to base, battery
Magma	0	supervised, return to base, battery
Rogma	50	supervised, return to base, battery
Treau	0	supervised, return to base, battery
Herma	0	supervised, return to base, battery

Table 3. Results with counter-measure sent by the authority dynamics controller

6. CONCLUSION

In the context of a mission operated jointly by an artificial agent and a human agent, we have presented an authority dynamics controller, based on a dependency graph between resources that can be controlled by the two agents. It is aimed at adapting the artificial agent or the human agent's behaviour in case of authority conflict on these resources. We have defined the relative authority between two agents relatively to a given resource through the properties of access, preemption, and control guarantee. The authority of an agent over a resource can thus change during the mission, in order to adjust its behaviour to the other agent's behaviour. Future experiments should allow us to study more precisely the criteria to justify an autority loss or gain for an agent over one or several resources, which we call *meta-authority*. The experimental aspect will be crucial because allowing an artificial agent to take over resources controlled by a human agent brings a lot of new challenges: agents'cohesion, communication and support of the human operator'situation awareness, operator's acceptance. Solutions allowing to influence the operator's actions without disturbing her (e.g. "subliminal" guidance, actions on the operator's situation awareness using counter-measures, etc.) must be developed.

REFERENCES

Bradshaw, J., Sierhuis, M., Acquisti, A., Feltovich, R., Hoffman, R., Jeffers, R., Prescott, D., Suri, N., Uszok, A., and Van Hoof, R. (2003). Adjustable autonomy and human-agent teamwork in practice: An interim report on space application. In Hexmoor et al. (2003), chapter 11.

Castelfranchi, C. and Falcone, R. (2003). From automaticity to autonomy: the frontier of artificial agents. In Hexmoor et al. (2003), chapter 6.

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

Dehais, F., Tessier, C., and Chaudron, L. (2003). Ghost : experimenting countermeasures for conflicts in the pilots activity. In *IJCAI03 Conference*. Acapulco, Mexico.

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

Fong, T., Thorpe, C., and Baur, C. (2002). Collaboration, dialogue and human-robot interaction. In *10th International Symposium on Robotics Research*. Lorne, Victoria, Australia.

Goodrich, M., McLain, T., Crandall, J., Anderson, J., and Sun, J. (2007). Managing autonomy in robot teams: Observations from four experiments. In *ACM/IEEE international conference on Human-robot interaction*. Washington, DC.

Goodrich, M., Olsen, D., Crandall, J., and Palmer, T. (2001). Experiments in adjustable autonomy. In *IJCAI'01 Workshop on Autonomy, Delegation and Control: interacting with autonomous agents*. Seattle, WA.

Hardin, B. and Goodrich, M. (2009). On using mixed-initiative control: a perspective for managing large-scale robotic teams. In *HRI'09, 4th ACM/IEEE Conference on Human-Robot Interaction*. San Diego, CA.

Hasslacher, B. and Tilden, M.W. (1996). Ieee ws on biomechatronics. In *Living Machines*. Minneapolis, MN.

Hexmoor, H., Castelfranchi, C., and Falcone, R. (2003). *Agent Autonomy*. Kluwer Academic Publishers.

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

Kortenkamp, D., Bonasso, P., Ryan, D., and Schreckenghost, D. (1997). Traded control with autonomous robots as mixed initiative interaction. In *AAAI-97 Spring Symposium on Mixed Initiative Interaction*. Stanford University, CA.

Mercier, S., Tessier, C., and Dehais, F. (2009). Adjustable autonomy without levels. In *NATO-SCI202 Symposium on "Intelligent uninhabited vehicle guidance systems"*. Neubiberg, Germany.

Nau, D., Muoz-Avila, H., Cao, Y., Lotem, A., and Mitchell, S. (2001). Total-order planning with partially ordered subtasks. In *IJCAI-2001*. Seattle, WA.

Scerri, P., Pynadath, D., and Tambe, M. (2003). Adjustable autonomy for the real world. In Hexmoor et al. (2003), chapter 10.

Schurr, N., Marecki, J., and Tambe, M. (2009). Improving adjustable autonomy strategies for time-critical domains. In *AAMAS 2009*.

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

Sheridan, T. and Verplank, W. (1978). Human and computer control of undersea teleoperators. Technical report, MIT Man-Machine Systems Laboratory, Cambridge, MA.

Wickens, C.D. (2008). Situation awareness: review of mica endsley's 1995 articles on situation awareness theory and measurement. *Human Factors*, 50(3).