Open access • Proceedings Article • DOI:10.1109/ICTAI.2012.90

Aerial: A Framework to Support Human Decision Making in a Constrained Environment — Source link

Pierre Yves Dumas, Amal El Fallah Seghrouchni, Patrick Taillibert

Published on: 07 Nov 2012 - International Conference on Tools with Artificial Intelligence

Topics: Contract Net Protocol

Aerial: a framework to support human decision making in a constrained environment

Pierre-Yves Dumas
THALES
Elencourt, France
pierre-yves.dumas-stag@fr.thalesgroup.com

Amal El Fallah Seghrouchni and Patrick Taillibert
LIP6
Paris, France
Amal.Elfallah@lip6.fr

***Abstract*—In this paper, we propose an architecture that uses a tender protocol, the Contract Net Protocol (CNP), to let human operators express their consent about the allocation of goals to Unmanned Aerial Vehicles (UAVs) in a constrained environment. The CNP has several good points: it has an appropriate level of automation; it is simple; it spares the bandwidth. But if bids are evaluated solely on the base of a numerical payoff, the CNP cannot fully convey human preference in complex situations. Thus, we extended and implemented the CNP within our framework, Aerial, to enable a more subjective human feedback. We detail how we build the bids and how we filter them to not flood the user. We also explain how we enable a dependable commitment in a dynamic context when the award time is not accurately foreseen.**

***Index Terms*—MAS, CNP, commitment, elicitation**

Increasing levels of autonomy in Unmanned Aerial Vehicles (UAVs) are expected to reduce the need for human intervention in operations. However, UAVs are not a substitute for human involvement in the battle-space. We focus on low levels of automation (LOA) because human control of UAVs is axiomatic for military relevance[1]. These low levels of automation state that the automatic systems make suggestions and carry them out when humans agree.

We believe that a tender protocol with human operators as awarder agents is suitable in that context. An early tender protocol is the Contract Net Protocol (CNP). It was primarily intended to handle fully automated negotiations, e.g. in e-commerce[2]. Usually, awarder agents are easily automated because they merely award the bid with the best score. But in real-life tenders between companies, the choice has two factors: the cost and the quality. We want to use human operators as awarder agents to identify the best couple plan/score.

Humans are not that rational that they could fully express their policy, which could be automated and convey human touch at full speed. We want to give human operators an opportunity to clarify their expectations when their policy may be misleading. This is elicitation. Each bid is meant to be an alternative way to understand what matters the most. For instance, if a default efficiency function allows to cancel *either x or y*, a bid may suggest to cancel explicitly x and another y.

We aim to identify in the default efficiency function what approximation could be critically given too much credit. For instance, as in the example above, two tasks a priori rated as of the same priority may seem more different once one out of the two has to be cancelled.

Our implementation and the choices we made are heavily dependent on our context, but our approach may be of some interest in many fields involving humans within mostly automated systems.

In section I we present the related work. In section II we detail how we build the bids and how we filter them to not flood the user. In section III we explain how we enable a dependable commitment in a dynamic context when the award time is not accurately foreseen. In section IV we expose our conclusion.

I. Related work

A. The Contract Net Protocol

TRACONET[3] is a system that manages a fleet of vehicles. Our context is alike because UAVs are vehicles. Agents in TRACONET are self-interested and must make negotiation decisions in real-time with bounded computational resources. Human operators can be considered to some extent as such agents.

Within TRACONET, Sandholm implemented the Contract Net Protocol (CNP), a tender protocol. The CNP was created by Smith[2] and Sandholm extended it[4].

The CNP involves an awarder agent and several bidder agents. It works as follows (figure 1 depicts the flow in the fashion standardized by the Foundation for Intelligent Physical Agents).

1) The awarder makes a task announcement.
2) The available bidders evaluate the task announcement and submit bids if they are suited.
3) The manager evaluates the bids and awards one.

If awarder agents are human operators, the CNP suits well the low levels of automation proposed by Sheridan and Verplank[5] and their counterpart in other LOA views[6][7]. These low levels of automation work as follows.

1) Automatic systems compute alternatives and narrow a selection.
2) Human operators choose one of these alternatives.
3) Automatic systems carry out the chosen alternative.

There are two motives to favor the CNP: it spares the bandwidth and it is intelligible.

- Our available bandwidth is scarce and the CNP has no extraneous message traffic. The control of the UAVs must be adequately addressed in the current air traffic management framework. Thus, it must bear the shortage of VHF frequency bands and limit itself to sparse and short messages[8]. Compared with other auctions (*e.g.* English auctions or Dutch auctions that are open ascending or descending price auctions), the tender is a first-price sealed bid auction that needs few messages. Furthermore, messages can be kept short and to the point through the use of the bid specification mechanism.
- Another good point of the CNP is its intelligibility because it is turn-based with only four turns: the announcing turn (awarder's first turn); the bidding turn (bidders' first turn); the awarding turn (awarder's second turn); the performing turn (bidders' second turn). However, to remain user friendly, the CNP must also convey not too many bids (see section II) and result in deterministic outcomes (see section III).

Usually, the CNP handles fully automated negotiations, thus we discuss next how to involve human operators.

B. Human involvement

Human operators in current architectures to control UAVs are the primary responsible factor in terms of goal-driven decision-making: they specify the constraints and demands settings for the automated systems (7.1.1 in [1]).

Since goals and constraints are the primitives that human operators use to interact with the automated system, a straight forward design is to base the automated systems on constraints. Playbook[9], the leading framework to manage UAV missions, has an architecture that revolves around a constraint-based planner (a modified SHOP2 [10]).

Our framework, Aerial, is also based on a planner: Airplan[11]. This planner is not constraint-based but it can handle constraints of three types:

- those preloaded as the mission context (*e.g.* winds that can vary over time, obstacles, ...)

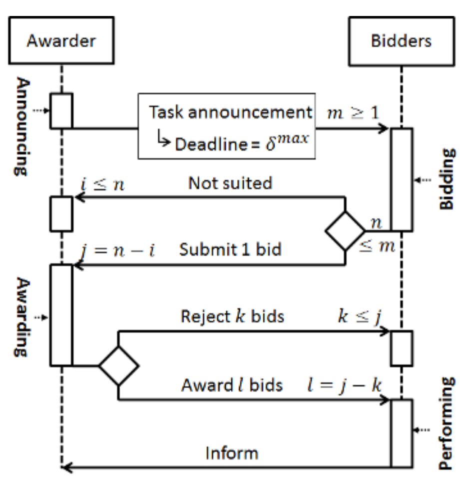

Fig. 1. Early CNP.

- those explicitly expressed by human operators (*e.g.* where to go, when, ...)
- those implicitly expressed by human operators (*i.e.* automatically deduced after what expressed human operators).

It is critical to be able to deduce some implicit constraints, otherwise the user may fail to provide the system with proper input. Miller *et al.*[9] give the following example. *Suppose an operator wants to perform a portion of Time Sensitive Targeting task, say, performing surveillance on a target, beginning at a specific time, and for a particular duration. The automation may check the availability of resources and find that no UAVs are available at the specified start time. If the automation has no additional models of the operators intent, it may stop searching, and report that such a plan cannot be created. A human subordinate, however, may understand that surveillance for some period is better than no surveillance at all, realize that adequate resources will become available 5 minutes later than the specified start time, and offer a relaxed plan as an alternative.*

In Playbook this issue is addressed by structuring the architecture so that such knowledge can be incorporated into the automation. This knowledge is abstracted from the task models, so the operator does not see it. Playbook offers the ability to provide priorities between alternate goals and states. These abstract value statements that a supervisor might provide are referred to as his or her policy for performance in the domain. A policy statement is an abstract, general, a priori statement of the relative importance or value of a goal state in the domain. In its simplest form, policy provides a method for human operators to mathematically define what constitutes efficiency.

Much like Playbook, our strategy is to propose high level features with a default setting. For instance, in the example above, a solution is to maximize a score based on a diminishing marginal utility function. A tick box allows the user to activate this feature and relax the problem. The default setting of this feature assumes that every additional minute is worth half as much as the previous one. Beyond a threshold, any more minute is worthless. The value of the first minute depends on the priority (low/medium/high/top) of the task. As in Playbook, human operators mathematically define what constitutes efficiency.

C. Elicitation

The limit of the approach above is that the scores behind these high level features can combine into an abstruse global score. If this global score is based on appreciations that are uncertain and approximate rather than fixed and exact, the mathematical efficiency ends up being disconnected of the reality.

In the end, users can be deceived by the very policy they expressed. This undermines the reason why humans remain in the system. Consideration of the technological viability of UAV systems, and the legal constraints, suggests that a human-in-the-loop system will be the most valuable and therefore the most likely mode of operation to provide the required supervision and discrimination[12]. Context sensitivity is important for assessing the quality of military decision making[13]. Humans have making capability not easily matched by artificial intelligence in computers. But without appropriate feedback, operators are indeed out-of-the-loop.

With the CNP, we aim to let the user regain some control when the global score gets too complex in terms of meaning. We think about bids as questions/suggestions like *Did you mean [this] and consider [that]? Because if so, the best solution is indeed ....*

Like Lily Rachmawati and Dipti Srinivasan[14], we do not trust real-valued parameters and prefer managing Pareto-optimal fronts. Our elicitation algorithm orders the fronts and converts the choices into bids of the CNP.

II. Another way to use the CNP

A. Considering quality/score rather than score alone

We will consider three examples to set up the context.

1) If several UAVs were ticked as last resort: Let two UAVs be d_1 and d_2. They are the only ones able to achieve $x : visit\ this\ city$. For some reason, they where ticked as last resort to achieve x. It could be because they are poorly suitable for the task; or because they are more expensive and because the task is dangerous; or because they belong to another team. The UAVs can be poorly suitable for a task because their relevant sensors or weapons are not the best ones in that situation; or because they are more difficult to control; or because they are less stealth. The reasons why d_1 and d_2 where ticked as last resort are different and it is difficult to compare them. But this does not mean that they are equal and that minor factors, like the fuel, should make the difference.

2) If several tasks were ticked to be done as soon as possible: Let two tasks be $x : visit\ this\ city$ and $y : visit\ this\ city$ as in figure 2. The wind blows west. There is three basic ways to understand the purpose of the user: achieve x as soon as possible, then achieve y as soon as possible (we note $\Downarrow x, y$); conversely, achieve y as soon as possible, then achieve x as soon as possible; or achieve x and y in any order as soon as possible (we note $\Downarrow (y + x)$). There are two possible plans, the first is to visit x at 1 and y at 9 (we note $\Downarrow x = 1; \Downarrow y = 9; \Downarrow (x + y) = 9$), the second is to visit y at 4 and x at 9 (we note $\Downarrow y = 4; \Downarrow x = 6; \Downarrow (x + y) = 6$). This calls for elicitation. But it would not be enought to just clarify the priority between x and y, since this priority may be balanced with the performance it allows. Hence, if asked what would be best out of $\Downarrow x, y$, $\Downarrow y, x$ and $\Downarrow (y + x)$, the user may prefer $\Downarrow x, y$, but if asked what would be best out of $\Downarrow x = 1; \Downarrow y = 9; \Downarrow (x + y) = 9$ and $\Downarrow y = 4; \Downarrow x = 6; \Downarrow (x + y) = 6$, the user may indeed prefer the latter. Neither the score alone nor the plan alone is enough to decide, both must be consider at once in a quality/score ratio.

3) If one task out of several ones must be canceled: Let three tasks be $x, y, z : visit\ these\ cities$ as in figure 3. There is no possible plan if no task is cancelled. All the tasks have

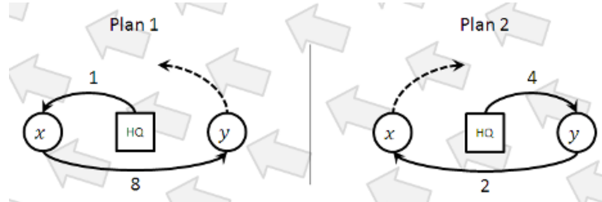

Fig. 2. If several tasks were ticked to be done as soon as possible.

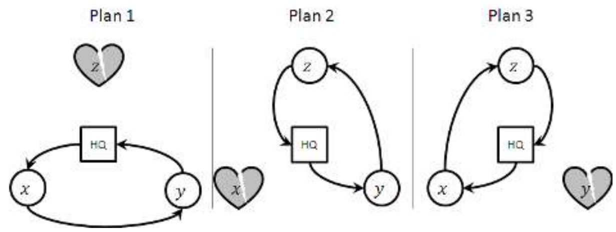

Fig. 3. If one task among several must be canceled.

the same priority. If any task is cancelled, the two other can be achieved. The first plan is to cancel z (we note $\neg z$), another is to cancel x, the last one is to cancel y. It may be hazardous to rely on a policy based on the assumption that x, y and z have exactly the same priority. Elicitation based on a quality/score ratio is safer.

In a real situation, the options introduced in the examples above tend to combine exponentially. The user cannot handle that many options and only the most relevant ones can be submited to his or her appreciation. We discuss next how to do that.

B. Generating the bids

Our generic approach works as follows:

1) Create options associated with tasks.
2) Decide how these options combine into a global score.
3) Decide what parts of the combination are suspicious.
4) Decide how to build alternatives revolving around the elicitation of these suspicious parts.
5) Propose these alternatives as bids.

Previous examples are based on options that can be associated with a task. These options are summarized in table I. A task has a priority which is either 1 (low) or 2 (medium) or 3 (high) or 4 (top).

With these options, we define a mathematical efficiency. We have rules to combine these options. For instance if two tasks have option 1 ticked, we minimize the earliest date at which they are both complete. We consider some of these choices as suspicious and we raise warnings whenever we use them. Similarly, if one task is cancelled rather than another one with the same priority, we also raise warnings. The order in which we raise warnings is summarized next.

- two tasks with option 1.
- two tasks with priority 3, one with option 2, one with option 3.
- two tasks with priority 3 with option 2.

TABLE I
OPTIONS ASSOCIATED WITH A TASK

☐ This task has to be done as soon as possible (option 1). Because of this option, this task has a top priority and cannot be canceled without explicit consent through the award of a bid. Without this option, no task can have a top priority. ☐ Can cancel any task with high priority? The default value is no. ⊡ Can cancel any task with medium priority? There is no default value, the user has to choose explicitly. ⊠ Can cancel any task with low priority? The default value is yes.
☐ This task has to last as much as possible (option 2). Due to diminishing marginal utility, we assume that no marginal utility can be worth cancelling a task with high or top priority. ⊠ Can cancel any task with a lower priority? The default value is yes.
☐ This task is better not performed by [*some UAVs*] (option 3). ⊠ Can cancel any task with a lower priority? The default value is yes.

- two tasks with priority 3 with option 3.
- two tasks with priority 3, one cancelled, cancelling the other one would achieve a similar result.
- two tasks with priority 2, one with option 2, one with option 3 (*i.e.* same warning than earlier but with priority 2 instead of priority 3, from now on we loop down to priority 1).
- ...
- two tasks with priority 1, one cancelled, cancelling the other one would achieve a similar result.

When we raise a warning, we may create bids. The number of bids is up to the user. He or she can change anytime this number. Each warning has a method to create bids. The number of bids that a method creates is easily known. If this number is less than twice the remaining number of available bids, all the bids for this method are computed at once. Else we compute no more bids. This way, we do not flood the user with bids.

If the user is interested in a specific warning, we compute the relevant bids. This is an optional step in our CNP that is not in the original CNP. This step can be repeated.

The methods are summarized next.

- n tasks $t_1, \ldots, t_n$ with option 1: n bids. The first one $\Downarrow t_1, (t_2 + \ldots + t_n)$ does t_1 as soon as possible and minimizes the earliest date at which all the other tasks are complete. The last bid is $\Downarrow t_n, (t_1 + \ldots + t_{n-1})$.
- $n + m$ tasks with priority k, n with option 2, m with option 3: m bids. All of them minimize how short is the shortest task among $t_1, \ldots, t_n$, because of diminishing marginal utility. Initially, the tasks must be normalized if they do not have the same norm (*e.g.* if one usually lasts for minutes and another one usually lasts for tens of minutes). Each bid relaxes the option 3 for a task among $t_{n+1}, \ldots, t_m$ to see how it improves the way tasks with option 2 are done. The user can request bids that consider several times the relax of the option 3.
- n tasks with priority k with option 2: n bids. Each maximizes the length of a task and minimizes how short is the shortest other one, because of diminishing marginal utility.
- n tasks with priority k with option 3: n bids. Each bid relaxes the option 3 for a task among $t_1, \ldots, t_n$ to see how it improves the rest. This improvement is the marginal cost. The bid is meant to decide if for one task, this option 3 is worth its marginal cost. The user can request bids that consider several times the relax of the option 3.
- n tasks with priority k, some must be cancelled: considering the lattice of the combinations to cancel these n tasks, one bid per feasible node not strictly dominated by another node. If there are too many nodes, only those that cancel up to two tasks are considered. If there is still too many nodes, only those that cancel a single task are considered.

Due to this methods, each bid is based on a local mathematical efficiency, as opposed to a global mathematical efficiency.

The automated systems stay in charge of most of the choices: bids cover only a few options among many that could be considered. But thanks to the ordered warnings, this few options are a priori the most significant. They allow the user to consider several alternatives based on the hypothesis that tasks that where roughly rated equal may not be equal any longer when they compete for resources.

This process to generate bids is part of our CNP that is part of Aerial. We detail next how this three layers interact.

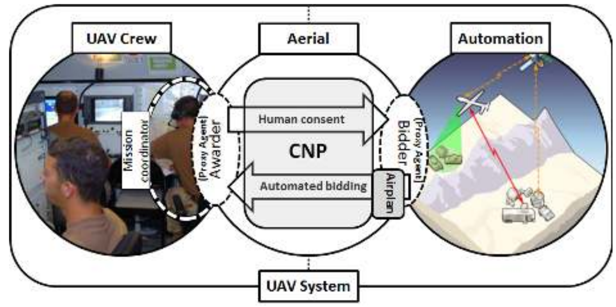

Fig. 4. Aerial within a UAV system.

C. Our architecture

A top view of a system that involves humans and computers leads to see on the one hand every one and on the over hand everything. Our CNP involves mainly two proxy agents, one that acts as a representative of every one, and one that acts as a representative of everything. We add a third agent to support the talk. Figure 4 details how the agents interact.

1) The Awarder: The Awarder conveys human consent. It acts as a representative of every human involved in allocating goals: a sensor operator, a flight operator, a mission coordinator, sometimes people outside the Control Ground Station (CGS).

2) The Bidder: The Bidder conveys automated bidding. It acts as a representative of everything that is automated in the UAV System, *i.e.* it is everything except the Awarder. Unlike

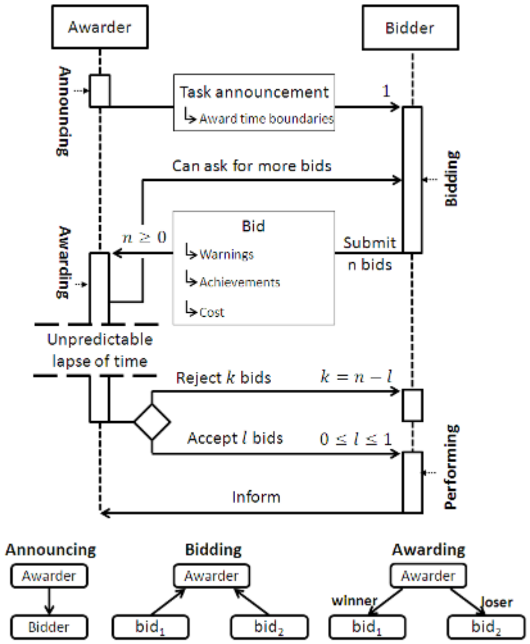

Fig. 5. Our human friendly Contract Net Protocol.

the contractors of the original CNP that are expected to be many, with up to a bid each, the contractor is alone but can submit many bids.

3) The Mediator: The Mediator supports the talk: it manages communications and provides minor features. For example it can early filter the UAVs that cannot achieve a task because they do not have the matching equipment.

These agents interact through our CNP as in figure 5. We consider that our CNP is human friendly unlike the original one meant to handle fully automated relationships. Its specificities are as follows.

- There is a single bidder that emits several bids rather than several bidders that emit a bid each.
- The call for proposal includes boundaries for the award time, this is meant to relax the use of a strict deadline.
- The awarding step lasts for an unforeseen lapse of time. It makes more difficult to get a dependable commitment.
- There are loser messages to free agents from their commitment, like in TRACONET.
- An optional step allows the awarder to request more bids.
- The bid includes three parts and does not include the plan.
  - The first part of the bid are the warnings about which choices are made by the local mathematical efficiency. These warnings are about the same in every bid, so they are factorized. The warnings that are specific, mostly those that tell what tasks are canceled, are enlighten. This warnings about the canceled tasks can be used to browse through bids, to find those who cancel other tasks and those who cancel the same ones.
  - The second part is the achievements that tell what was optimized. The outcome may depend on the award time. This outcome is what the commitment is meant to ensure.
  - The third part is a score, or conversely a cost, that is the local mathematical efficiency. This may depend on the award time as well. The decreasing of the score, or the increasing of the cost, is the cost of the commitment.

Most of these features are related to the commitment. What is the commitment and how it is computed is discussed in the next section.

III. Commitment

In this section, we will refer to what follows.

D	$= \{d_1, \ldots, d_m\}$ a set of m UAVs.
g_0	a set of physical constraints.
G	$= \{g_1, \ldots, g_n\}$ a set of n goals where every goal is a set of tactical constraints.
A	a Boolean matrix $m \times n$ where $a_{i,j}$ is true if g_j is allocated to d_i and false otherwise.
σ_i	$= g_0 \wedge (\neg a_{i,1} \vee g_1) \wedge \ldots \wedge (\neg a_{i,n} \vee g_n)$ the summed up constraint on the behavior of d_i. The set of physical constraints g_0 always applies. If $a_{i,j}$ is true, then this states that g_j is allocated to d_i and the set of tactical constraints g_j applies.

A UAV d_i carries out a plan p_i computed by Airplan minimizing a cost c_i. Thus $Airplan(d_i, \sigma_i) = (p_i, c_i)$. A plan p_i is a set of constraints, thus it is also a single complex constraint (the conjunction of the constraints it includes as a set).

The new goal is g_{n+1}. As every goal, g_{n+1} is a set of constraints and as such can be seen as a single complex constraint.

The deadline is δ^{max}. The commitment is to consider that this plan p_i constrains the behavior of d_i till δ^{max}. Thus, at any given time t, the commitment is a constraint $(t > \delta^{max}) \vee p_i$.

As three constraints, g_{n+1}, σ_i and $t > \delta^{max} \vee p_i$ can be summed up in a single constraint $g_{n+1} \wedge \sigma_i \wedge ((t > \delta^{max}) \vee p_i)$. This constraint is the new input that Airplan needs to compute a new plan p'_i with a cost c'_i that allows the UAV d_i to success in these three points:

- to achieve the new goal, thanks to g_{n+1}.
- to achieve the other goals it was having yet, thanks to σ_i.
- to be faithful to its commitment till the deadline of the tender δ^{max}, thanks to $(t > \delta^{max}) \vee p_i$.

$$Airplan(d_i, g_{n+1} \wedge \sigma_i \wedge ((t \leq \delta^{max}) \Rightarrow p_i)) = (p'_i, c'_i)$$

The marginal cost to carry out a new goal is $mcost(d_i, g_{n+1}, \delta^{max}) = c'_i - c_i$. If the UAV and the new goal are left implicit:

$$mcost(\delta^{max}) = c'_i - c_i$$

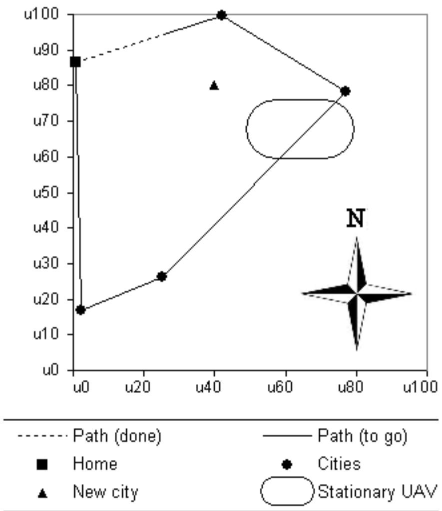

Fig. 6. Map. With a constant speed v, one space unit u is the distance a UAV can cover in one time unit u of the chart 7.

A. Overview

Let d_1 and d_2 be two vehicles, d_1 makes a round trip, d_2 is stationary. Their trajectories are as in figure 6. The place is an empty Euclidean space. d_1 has to visit four cities. The cities can be visited anytime in any order. Every trajectory is computed by a constraint-based Traveler Salesman Problem (TSP) solver. Now, a new city has to be visited. Any vehicle could technically do it at the cost of an additional distance. But there is a risk that cannot be mathematically expressed and that requires human consent. This risk is subjective, it could be friendly fire, collateral damage, failure or casualties. This risk may be related to the vehicle, thus it makes sense to neither exclude d_1 nor d_2, even if d_2 is most of the time closer. The decision is complex and human operators need some time to make it. How long it will take is unknown.

The decision making is stressful, thus human operators expect the awarded UAV – if any – to carry out the mission when requested to do so. They want *commitment*. Bellow is how commitment is described by Sandholm[4]: *In mutual negotiations,* commitment *means that one agent binds itself to a potential contract while waiting for the other agent to either accept or reject its offer. [...] When accepting, the second party is sure that the contract will be made, but the first party has to commit before it is sure.* The commitment starts when one UAV bids and lasts till the bid of this UAV is awarded or rejected.

The original CNP uses an expiration time as part of the task announcement. Smith says about it that it is a deadline for receiving bids and that time is not critical in the negotiation process. For example, bids received after the expiration time of a task announcement are not catastrophic: at worst, they may result in a suboptimal selection of contractors. However time is indeed critical and this deadline is a way to handle it.

Suppose that the environment is deterministic, as in the above example since the trajectories of the UAVs are known. Let δ^{max} be this deadline. We know where each UAV will be at δ^{max}. Thus one bid can be computed considering where one UAV will be rather than where the UAV is, this is simply anticipation. Let be δ the award time. If $\delta \leq \delta^{max}$, no matter where the UAV is, it can still continue to perform its initial plan till δ^{max} and then carry out the task as expected. Otherwise, if $\delta > \delta^{max}$, it is safer to cancel the tender because of uncertainty. Thus, the deadline, in a deterministic environment, is a way to enable a trustworthy commitment.

We do not consider that the environment is stochastic because we believe that its complexity is beyond stochastic modeling. It cannot be as simple as generating targets according to a spatio-temporal Poisson process [15]. Most often, the changes of the environment are accurately predicted. This enables bounded rational decision making.

The commitment is a constraint over the behavior of a UAV: it may narrow how this UAV can participate in other tenders. Thus, it is better to free a UAV of its commitment as soon as possible. Sandholm uses explicit loser messages to do that and do not let the contractors wait till the deadline. We do as well. When one bid is awarded, all other are automatically rejected. One bid may also be early rejected when no bid has yet been awarded.

To motivate the Awarder to respond quickly, Sandholm[4] proposes that a bid includes a required payment fn (element 6.5.b of its *contractee message*) that increases over time. The later is the award time δ, the more expensive is the required payment. Unlike the expiration time, fn does not force a strict deadline, which can inefficiently constrain the Awarder's deliberation scheduling as we explain next.

B. The Awarder's deliberation scheduling

The Awarder needs time to decide what bid to award, if any. This decision is complex and it is not possible to foresee during the task announcement how long it takes to make it. Further vicissitudes may delay it:

- Humans may be busy due to a workload pick in another process.
- Many or few bids may be received.
- Bids may be surprisingly simple (*e.g.* operators may find obvious the prevalence of a bid over others when computers saw them all as Pareto dominant).
- Bids may be surprisingly complex (*e.g.* operators may need additional information).
- Communications may lack continuity (*e.g.* for stealthy reasons).
- Computers may be busy (*e.g.* it happens during fast paced simulation).

Thus, the spontaneous deliberation scheduling is unpredictable. We may however decide to force a strict deadline upon it. It would be convenient since this deadline, in our deterministic environment, makes possible the commitment. Let δ be the end of the spontaneously scheduled deliberation is unpredictable and the award time. There are three possible outcomes.

1) $\delta > \delta^{max}$: The commitment is over and without any further information, it is best to cancel the tender. But sometimes the commitment could actually be extended, and if so, it is not efficient to cancel the tender. Let us focus on d_1 in our example. If $\delta^{max} = 125$ and if the spontaneously scheduled deliberation ends at 175, it would seem safer to cancel the whole process since the commitment is over. But $mcost(175) = mcost(125)$, so the commitment could be extended.

2) $\delta \not\approx \delta^{max} \wedge \delta < \delta^{max}$: The commitment is not over and the task can be successfully carried out. It is easy to get this situation with a high δ^{max}. However, the commitment is a constraint part of the input of Airplan. This constraints constrains more when δ^{max} is higher, thus, if δ^{max} is too high, no UAV can afford this commitment and no bid come. Even without being extreme, $\delta < \delta^{max}$ means that the problem may have been relaxed and that better plans may have been lost. Let us still focus on d_1 in our example with $\delta^{max} = 125$. If now the spontaneously scheduled deliberation ends at 75 then $mcost(d_1, g_{n+1}, 75) < mcost(d_1, g_{n+1}, 125)$. With the other UAV, $mcost(d_2, g_{n+1}, 75) = mcost(d_2, g_{n+1}, 125)$ but $mcost(d_2, g_{n+1}, 75) > mcost(d_1, g_{n+1}, 75)$ and $mcost(d_2, g_{n+1}, 125) < mcost(d_1, g_{n+1}, 125)$. It means that if humans have no subjective preference, d_1 will be chosen rather than d_2 when it would have been more efficient to choose d_2.

3) $\delta \approx \delta^{max} \wedge \delta < \delta^{max}$: The remaining situation has none of the disadvantages of the two previous ones, there is no need to extend the commitment and the problem could not have been relaxed. However, this situation should be unlikely since δ in till unpredictable because δ^{max} is decided during the task announcement. The complexity is beyond what could be stochastically modeled. Asking the operator would be unfair and useless(we know that δ^{max} cannot be well predicted), stressful (the operator understands that it has consequences) and counterproductive (giving the operator something more to worry about).

Indeed, Sandholm is right, a strict deadline can inefficiently constrain the Awarder's deliberation scheduling. The approach we propose it to let human operators be aware of the consequences of how much time they spend to decide. Rather than telling them what would be $mcost(\delta^{max})$ for a single deadline δ^{max}, we tell them what is $mcost(\delta)$ for date of end of the spontaneous deliberation. In our example, the result if figure 7.

It is similar to the required payment fn of Sandholm (element 6.5.b of its contractee message) meant to free the bidders from the commitment of their bids. What we want to focus on is the advantage for the human decision process.

C. $mcost(\delta)$ is intelligible

One may worry if the curve $mcost(\delta)$ does make the bid too complex and unintelligible. There are three reasons why $mcost(\delta)$ remain intelligible.

1) It is non-decreasing: The curve $mcost(\delta)$ is non-decreasing. This property is a straight consequence of how

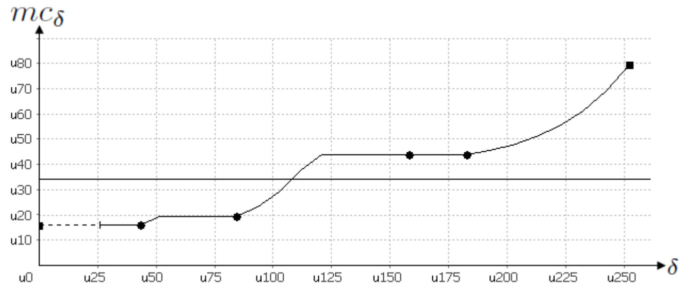

Fig. 7. $mcost(d_1, g_{n+1}, \delta)$ and $mcost(d_2, g_{n+1}, \delta)$. The flat one is $mcost(d_2, g_{n+1}, \delta)$. With a constant speed v, one time unit u is how long a UAV needs to cover one space unit u of the chart 6.

$mcost(\delta)$ was expressed: $mcost(\delta^{max}) = c_i' - c_i$ with $Airplan(d_i, g_{n+1} \wedge \sigma_i \wedge ((t \leq \delta^{max}) \Rightarrow p_i) = (p_i', c_i')$ minimizing c_i' and $Airplan(d_i, \sigma_i) = (p_i, c_i)$ minimizing c_i. Thus, when δ^{max} gets higher, $((t \leq \delta^{max}) \Rightarrow p_i)$ constrains more, c_i' is less minimized (or remains equal) and $mcost(\delta^{max}) = c_i' - c_i$ increases (or remains equal).

2) $[\delta^{min}, \delta^{max}]$ should be short, up to half an hour: It leaves few room to display many phases.

3) There are steps: A step is wherever $mcost(\delta)'$, the derivative of $mcost(\delta)$, is null. To have an idea of how common steps are, we randomly generated hundreds of missions like the one of d_1. The closer the UAV gets to home, the most unlikely is it to be in a step. To illustrate this, we split each mission into 20 equal sections. We use each mission twice, as in a round trip, so that the order of the TSP solution is neutral. We make the initial number of cities vary. For every section, we display the probability to be in a perfect step ($\alpha = 0$). Figure 8 shows the results for 5 and 12 initial cities. With more cities, there are more combinations available, thus the probability to quickly find a next step when leaving one is higher. This is why the probability to be in a step decreases slower with 12 cities than with 5. To put it another way, with only 5 cities, the probability is high in the sixteenth section to straightly fly back home. In that case, there are no more steps within the remaining sections. With 12 cities, in the sixteenth section, the probability to have to visit a few more cities is high, and thus there is still room for some steps within the remaining sections.

The above experiment use constraints that are too simple to be realistic. Fortunately, complex constraints seem to raise a bit more steps than basic constraints. Complex constraints tend to make opportunities and sanctions stronger. Thus, it is more likely to have to wait for long before a very attractive opportunity arises. It may sound like a bad situation, but instead it results in a very long and readable step. Our tests do not allow us to be very affirmative. As we said earlier, it is not an easy task to stochastically model missions. There are two major biases:

- What if the new goal can be postponed to a second trip? If so, there is a permanent step, as with d_2.
- What if the new goal cannot be postponed forever? If so, the probability to be in a step at the beginning of the trip

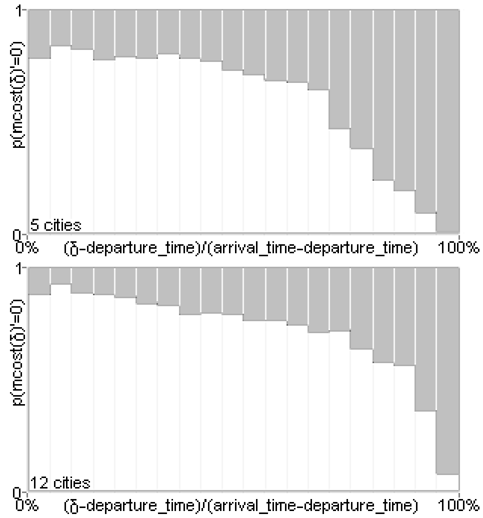

Fig. 8. Steps appear often in $mcost(\delta)$. Steps appear less often when the UAV is almost home. The number of visited cities has little incidence.

is not that high.

In the end, the probability to be in a step mainly depends on how long the new goal can be postponed. If the new goal is an emergency with an absolute priority, there is no step. But we address situations where time does not run that quickly and where humans can decide to take longer to come out with a decision.

D. How to compute $mcost(\delta)$

Since $mcost(\delta)$ is non-decreasing, we can use an imperfect planner that relies on a discretized space, that does not always find a solution (even when some exist), and that may not find the best solution. Planners that can handle the operational constraints used to plan UAV flights often have these limits. Airplan is no exception. We can use the increasing property to complete and flatten $mcost(\delta)$.

If mc_{δ_0} is known for $\delta = \delta_0$, and if mc_{δ_2} is known for $\delta = \delta_2$, and if $\delta_2 > \delta_1 > \delta_0$, then $mc_{\delta_2} > mc_{\delta_1} > mc_{\delta_0}$. This means that something can be known about mc_{δ_1} without computing it. If $mc_{\delta_2} - mc_{\delta_0} < \alpha$ where α is the least meaningful difference, this means that mc_{δ_1} does not even have to be computed. Same if $\delta_2 - \delta_0 < \beta$ where β is the least meaningful difference. We use that in some sort of root-finding algorithm:

Start with the ordered set of known costs $mc^{all} = \{mc_{\delta^{min}}, mc_{\delta^{max}}\}$ and the matching ordered set of possible award times $\delta^{all} = \{\delta^{min}, \delta^{max}\}$.
Select any consecutive couple (mc_i, mc_k) in mc^{all} such as $mc_k - mc_i \geq \alpha$ and $\delta_k - \delta_i \geq \beta$. A possible award time δ_j such as $\delta_i < \delta_j < \delta_k$ must also exists.
Repeat as long as a new couple is found.
Choose a possible award time δ_j halfway between δ_i and δ_k.
Add mc_{δ_j} to mc^{all} and δ_j to δ^{all}

IV. Conclusion

We have presented Aerial, a framework meant to let humans express their consent over the allocation of tasks to UAVs in a real-time context. Aerial is based on an extended CNP whom the bids aim at elicitation. A strong commitment is granted despite an unpredictable lapse of time between the call for proposals and the award time.

The decision remains complex. Bids include complex data like warnings and achievements, and scores may vary over time, till the expiration time. Considering these data, it is up to the human operators to decide what couple plan/score is the best. Human operators are given the opportunity to plainly and soundly control what the automated system carries out.

In future work we will address concurrency when several teams of human operators share a single fleet.

References

[1] NATO Research and Technology Organisation *Uninhabited Military Vehicles (UMVs): Human Factors Issues in Augmenting the Force* Final Report of the RTO Human Factors and Medicine Panel (HMF) Task Group HFM-078/TG-017 Jul. 2007.
[2] Reid G. Smith *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver* IEEE Transactions On Computers Vol. C-29, No. 12, Dec. 1980.
[3] Tuomas Sandholm *An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations* Eleventh National Conference on Artificial Intelligence pp. 256-262, Jan. 1993.
[4] Tuomas Sandholm and Victor Lesser *Issues in Automated Negotiation and Electronic Commerce - Extending the Contract Net Framework* Proceedings of the first International Conference on Multiagent Systems ICMAS 1995.
[5] Thomas B. Sheridan, W. L. Verplank *Human and computer control of undersea teleoperators* Technical Report, MIT Man-Machine Laboratory, Cambridge, MA pp. 26, Aug 1992.
[6] R.M. Taylor, S. Abdi, R. Dru-Drury and M.C. Bonner *Cognitive cockpit systems: information requirements analysis for pilot control of cockpit automation* Engineering psychology and cognitive ergonomics, Vol. 5, Aerospace and transportation systems Ch. 10, pp. 81-88, 2001.
[7] AGARD *Improved guidance and control automation at the man-machine interface* AGARD advisory report AR-228 1986.
[8] Andr J. Clot *Communications Command and Control - The crowded spectrum* Development and Operation of UAVs for Military and Civil Applications pp. 2B-1 - 2B-8, Apr. 2000.
[9] Christopher A. Miller et al. *The Playboo Approach to Adaptative Automation* Proceedings of the Human Factors and Ergonomics Society Annual Meeting Vol 49 No. 1 pp. 15-19, Sep. 2005.
[10] T.-C. Au, O. Ilghami, U. Kuter, D. Wu, F. Yaman, H. Munoz-Avila, J.W. Murdock *Applications of SHOP and SHOP2* Intelligent Systems, IEEE Vol. 20 No. 2 pp. 34-41, Mar. 2005.
[11] Michal Soulignac, Patrick Taillibert and Michel Rueher *Time-minimal Path Planning in Dynamic Current Fields* Proceedings of the International Conference on Robotics and Automation (ICRA) pp. 2473-2479, May 2009.
[12] P.D. Kennett *Autonomous Killing Machines The Technical, Legal and Moral Implications* Defence Research Paper, Advanced Command and Staff Course No 8, Mar. 2005.
[13] R.S. Bolia, W.T. Nelson, M.A. Vidulich, and R.M. Taylor *From chess to chancellorsville: Measuring decision quality in military commanders. In: D.A. Vincenzi, Mouloua, M. and Hancock, P.A.* Human Performance, Situation Awareness and Automation: Current Research and Trends (HPSAA II) Vol. I pp. 269-273, 2004.
[14] Lily Rachmawati, Dipti Srinivasan *Incorporating the notion of relative importance of objectives in evolutionary multiobjective optimization*, IEEE Vol. 14 No. 4 pp. 530-546, Aug. 2010.
[15] Emilio Frazzoli and Francesco Bullo Decentralized Algorithms for Vehicle Routing in a Stochastic Time-Varying Environment 43rd IEEE Conference on Decision and Control Dec. 2004.