Chrpa, Lukáš, Pinto, Jose, Ribeiro, Manuel, Py, Frederic, Sousa, Joao and Rajan, Kanna

On Mixed-Initative Planning and Control for Autonomous Underwater Vehicles

Original Citation

Chrpa, Lukáš, Pinto, Jose, Ribeiro, Manuel, Py, Frederic, Sousa, Joao and Rajan, Kanna (2015) On Mixed-Initative Planning and Control for Autonomous Underwater Vehicles. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015), 28th September - 2nd October 2015, Hamburg, Germany.

On Mixed-Initiative Planning and Control for Autonomous Underwater Vehicles

Lukáš Chrpa*, José Pinto†, Manuel A. Ribeiro†, Frédéric Py†, João Sousa†, Kanna Rajan†
*PARK Research Group, School of Computing and Engineering, University of Huddersfield, UK
†LSTS, Faculty of Engineering, University of Porto, Portugal
l.chrpa@hud.ac.uk,{zepinto,maribeiro,jtasso,kanna.rajan}@fe.up.pt, fredpy@gmail.com

***Abstract*— Supervision and control of Autonomous underwater vehicles (AUVs) has traditionally been focused on an operator determining a priori the sequence of waypoints of a single vehicle for a mission. As AUVs become more ubiquitous as a scientific tool, we envision the need for controlling multiple vehicles which would impose less cognitive burden on the operator with a more abstract form of human-in-the-loop control. Such mixed-initiative methods in goal-oriented commanding are new for the oceanographic domain and we describe the motivations and preliminary experiments with multiple vehicles operating simultaneously in the water, using a shore-based automated planner.**

I. Introduction

Autonomous Underwater Vehicles (AUVs) have made steady gains as tools for scientific observations and security applications in recent years. They have shown their utility in both benthic [1] and upper water column exploration [2], [3].Typically they have been deployed as single vehicles controlled by a single operator.

As these robots have become more affordable and robust, it is becoming viable to own and operate multiple vehicles, sometimes simultaneously. The LSTS laboratory, for instance, participates in an annual exercise where multi-vehicle operation and control are the key emphasis for security and upper water exploration [4]. These experiments target coordinated observations primarily to look for features such as frontal zones, blooms and plumes in the coastal ocean [5]. Networked heterogeneous robotic vehicles can be applied to complex scenarios where mobile sensing nodes can be scheduled according to their specific capabilities towards a common goal. However, in order for tasking vehicles appropriately, they must be aware of the state of the entire system and any vehicle-specific capabilities and parameters.

Our experiments in coordination and control, have revolved on the use of advanced decision-support tools in the LSTS toolchain [6] often with onboard machine intelligence to synthesize actions with continuous interleaved planning and execution. Our AUV platform is the Light AUV (LAUV) [7], an advanced and robust vehicle with an open source (and open) architecture (Fig. 1).

In these experiments, decision support, situational awareness, control, planning, data visualization and archiving is provided by software including NEPTUS, with DUNE providing low-level functional control and IMC the middleware message-passing mechanism. Task planning on NEPTUS involves either sending waypoints

Fig. 1. The human-portable LSTS Light Autonomous Underwater Vehicle (LAUV) [7] is a multi-use vehicle for benthic and upper water-column exploration.

to the AUV when on the surface, or more recently, sending abstract plans to the vehicle formulated as goals for onboard plan synthesis [4]. This has proved adequate for commanding a single vehicle. However, as we move towards mixed-mode multi-vehicle operations at sea, the need arises to use abstraction for goal formulation. We envision that a single NEPTUS user would rapidly plan a set of goals for a collection of vehicles in a mixed-initiative formulation [8], and then dispatch these goals for execution for AUVs. The goals would initially be targeted for distinct tasks on individual vehicles with the aim of obtaining field experience on such multi-vehicle operationand then gravitate towards coordinated experiments with temporal and task constraints involving complex task handoffs between AUVs and in the near future UAVs.

In this paper, we describe the first steps towards such a mixed-initiative planning and control tool-set that combines "high-level" goal-oriented mission planning and "low-level" control of the vehicles for AUVs only. In particular, the operator generates multiple tasks in NEPTUS which encodes them as a planning problem that is solved by a domain-independent planning engine and the plan is then dispatched to every vehicle as a sequence of tasks to perform. Control-loop closure occurs onboard; however task closure for now is done visually by the operator on NEPTUS who can observe the vehicle behavior in real-time using acoustic modems for communicating AUV states. We have evaluated our approach on a mine-hunting scenario with multiple AUVs with different payloads. To the best of our knowledge this is the first work in this domain, which use mixed-initiative automated planning and control methods on shore.

The paper is organized as follows. Section II situates this work in the context of previous efforts, Section III provides background material, with Section IV the core of the paper, highlighting the planning technique.

Results from experiments are highlighted in Section V and we conclude with future work in Section VI.

II. Related Work

In the oceanographic domain, command and control of a vehicle is typically waypoint based with incremental waypoint achievement as a means to identify goal achievement. Our previous work is among the few to include automated planning in this harsh domain where we have demonstrated embedded *continuous* planning and execution in the context of scientific upper water-column exploration on a single AUV [9], [10], [11]. Recent work [12] shows more interest along similar lines. Multi-vehicle planning in the oceanographic context has continued to revolve along waypoint based low-level control [13]. Multiple glider deployments [14] have also been demonstrated, again using simple waypoint-based methods. Further, our work is informed by past efforts in mixed-initiative command/control in the space domain; the MAPGEN system continues to command the rover *Opportunity* on the surface of Mars using automated planning methods [8] also used in this work. While we have demonstrated a form of mixed-initiative control on AUVs, this has been in the context of a single AUV while tracking gradients [11] and where automated decision-making was onboard the vehicle. To the best of our knowledge using automated planning as a means to provide abstraction in control over multiple vehicles in the oceanographic domain is novel.

III. Background & Motivation

LSTS, has developed a set of software tools to support the operation of heterogeneous vehicle networks [15]. Operators interact with vehicles via NEPTUS, a graphical decision-support system with visualization and analysis capabilities that allows users to view all incoming vehicle data in real-time, to define vehicle objectives and to supervise their execution.

Embedded on the vehicles, the DUNE executive manages localization, path-planning, data acquisition and command execution on different types of vehicles and other embedded systems like data loggers and communication gateways. IMC is the middle-ware used for conjoining all the sensory data with the various platforms. DUNE has an in-built executive which is capable of parsing and executing script-based plans. The language used for describing plans is graph-based where each node represents a parameterized behavior and transitions between nodes occur in response to events such as behavior termination or failures. Moreover, DUNE allows the specification of configuration parameters, like payload settings, on transitions. This simple plan specification has been used extensively for deterministic behaviors. In order to allow other types of behavior definition and execution, an API was devised that allows external modules to guide vehicles and control their payload configuration.

In this work, we describe a shore based automated planner which connects to the NEPTUS API for mixed-initiative command and control. Automated Planning

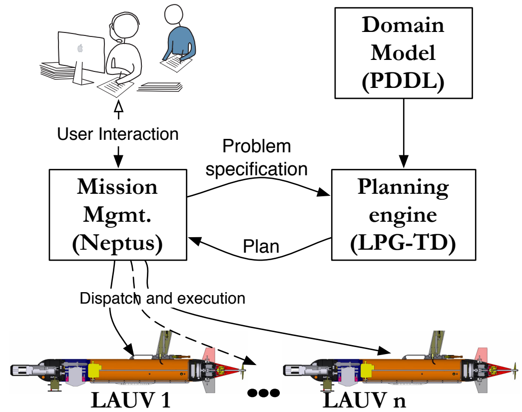


Fig. 2. A modular architecture of the system.

deals with a problem of finding a sequence of actions that transform the environment from a given initial state to a desired goal state [16]. The simplest form of automated planning works in deterministic, fully observable and static environment, where effects of actions are instantaneous. Temporal planning, which is the focus of our interest, allows "durative actions" whose planning and execution makes explicit use of, and reasons with, the notion of time.

We use PDDL 2.1 [17] for representing our planning problems. The environment is described by predicates and numeric functions. Actions are specified via pre-conditions, effects and the duration of their execution. Preconditions are sets of logical expressions that must be true in order to have the actions executable. In PDDL, these expressions can take place prior to starting action execution, prior to finishing action execution, or over the whole time period when the action is executed. Effects are sets of literals and function assignments that become true when the action is executed. In PDDL, effects can occur just after starting, or just after finishing action execution.

The PDDL representation allows us to specify domain models and problem specifications separately. Usually, one domain model is used for a class of problems. In particular, a domain model consists of predicate and function descriptions and action definitions, while a problem specification consists of definition of objects, an initial state and a set of goal conditions.

In domain-independent planning, planning engines and domain models are decoupled. A planning engine can deal with various domain models, and a domain model can be accepted by various planning engines. Therefore, if the domain model is modified, there is no reason to modify the planning engine; equally it is possible to replace one planning engine with another without changing the domain model. As shown in Fig. 2, a user specifies the mission in a control system like NEPTUS, which then automatically generates planning problem description that is accepted by the planning engine. Given the domain model and problem specification, the planning engine returns a plan (if one exists) to NEPTUS which in turn, distributes the plan among the vehicles, where it is executed. Since the planning engine is used as a "black-box" we have extended NEPTUS in order to generate problem specification in PDDL as well as to process PDDL-compliant plans.

Specifying problems in PDDL is relatively straightforward. For example, predicates are defined in the form of (at noptilus1 task1-loc) and function assignments are defined in form of (= (battery-use noptilus1) 50). A plan action is in the form 48.0: (MOVE XPLORE1 XPLORE1-DEPOT T01-LOC) [1365.000], where the first number refers to the time-stamp when the action is executed, while the second refers to the duration of action execution. Every action accepts a single vehicle as a parameter.

In this work we use **LPG-TD** [18] as the planning engine. **LPG-TD** is based on a local search in Planning Graphs [16] which allows executing actions that do not interfere with each other in each plan step. While it is a mature planner, most state-of-the-art planning engines either do not support required features (such as numeric functions or durative actions) nor do they scale well. For instance, while both **Optic** [19] and **EUROPA**$_2$ [20] offer richer representations, our experimentation with these demonstrated that their performance did not scale well for problems we envision for such mixed-initiative interaction.

IV. Goal-oriented Mission Planning

In short, automated planning is used to decide what tasks, which are specified by a user, are performed by which AUV and when. Various constraints are considered (e.g. battery limits).

We have conceptualized mission requirements in the form of a domain model specification. This conceptualization is divided into three categories: object types, predicates and functions, and actions similar to work of Shah et al. [21]. Object types refer to classes of objects that are relevant for the planning process such as: *vehicle* (V), *payload* (P), *phenomenon* (X), *task* (T), *location* (L). By "phenomenon" we mean a target object or area of interest, where we assume that such phenomena are deterministic and static. Tasks are considered as atomic, to be fulfilled by a single vehicle.

Predicates and functions describe states of the environment. In particular, predicates represent relationships between objects, and functions refer to quantity of resources related to the objects. In our case, we have defined: $at \subseteq V \times L$ – a location of the vehicle, $base \subseteq V \times L$ – a location of the vehicle's *depot*, i.e. known positions where a vehicle is expected to be safe when on the surface, $has \subseteq V \times P$ – whether a payload is attached to the vehicle, $at\text{-}phen \subseteq X \times L$ – a location of the phenomenon, $task \subseteq T \times X \times P$ – which describes a task of getting data about a phenomenon from a specific payload, $sampled \subseteq T \times V$ – whether data of a given task has been acquired by the vehicle, $data \subseteq T$ – whether the task data has been acquired from the vehicle, $dist : L \times L \rightarrow \mathbb{R}^+$ – a distance between two locations, $speed : V \rightarrow \mathbb{R}^+$ – speed over ground of the vehicle, $battery\text{-}level : V \rightarrow \mathbb{R}_0^+$ – the amount of energy in a vehicle's battery, $battery\text{-}use : V \cup P \rightarrow \mathbb{R}^+$ – battery consumption per distance unit (moving a vehicle) or per time unit (using a payload). We currently make an assumption of linear energy use both for moving or using a payload.

Actions when executed modify the environment according to their effects. We have specified 4 actions (we denote t_s as time when an action is executed, and t_e as time when an action execution ends):

- *move*(v, l_1, l_2) – the vehicle v moves from its location of origin l_1 to its desination location l_2. As a precondition is must hold that in t_s: $(v, l_1) \in at$, $battery\text{-}level(v) \geq dist(l_1, l_2) * battery\text{-}use(v)$; and in t_e: $\neg\exists v_x \neq v : (v_x, l_2) \in at$. The effect is that in t_s: $(v, l_1) \notin at$, and $battery\text{-}level(v) = battery\text{-}level(v) - dist(l_1, l_2) * battery\text{-}use(v)$, and in t_e: $(v, l_2) \in at$
- *sample*(v, t, x, p, l) – the vehicle v samples a phenomenon x by the payload p. As a precondition it must hold that in t_s: $battery\text{-}level(v) \geq (t_e - t_s) * battery\text{-}use(p)$, and in $[t_s, t_e]$: $(v, l) \in at$, $(x, l) \in at\text{-}phen$, $(v, p) \in has$ and $(t, x, v) \in task$. The effect is that in t_s: $battery\text{-}level(v) = battery\text{-}level(v) - (t_e - t_s) * battery\text{-}use(p)$, and in t_e: $(t, v) \in sampled$.
- *survey*(v, t, x, p, l_1, l_2) – the vehicle v surveys the area (between l_1 and l_2) of a phenomenon x occurrence by the payload p. As a precondition is must hold that in t_s: $(v, l_1) \in at$, $battery\text{-}level(v) \geq dist(l_1, l_2) * battery\text{-}use(v) + (t_e - t_s) * battery\text{-}use(p)$, and in $[t_s, t_e]$: $(x, l_1) \in at\text{-}phen$, $(x, l_2) \in at\text{-}phen$, $(v, p) \in has$ and $(t, x, v) \in task$. Also, no other vehicle can perform the survey action over the phenomenon x in $[t_s, t_e]$. The effect is that in t_s: $(v, l_1) \notin at$, and $battery\text{-}level(v) = battery\text{-}level(v) - (dist(l_1, l_2) * battery\text{-}use(v) + (t_e - t_s) * battery\text{-}use(p))$, and in t_e: $(v, l_2) \in at$, $(t, v) \in sampled$.
- *collect-data*(v, t, l) – the data associated with a task t is collected by vehicle v. As a precondition it must hold that in $[t_s, t_e]$: $(v, l) \in at$, $(v, l) \in base$ and $(t, v) \in sampled$. The effect is that in t_e: $t \in data$.

As an example, the Sample action is encoded as depicted in Fig. 3.

The user specifies mission tasks in NEPTUS's front-end by pointing to the locations/area of the phenomena

```
(:durative-action sample
 :parameters (?v - vehicle ?l - location
    ?t -task ?o - phenomenon ?p - payload)
 :duration (= ?duration 60)
 :condition (and (over all (at-phen ?o ?l))
   (over all (task ?t ?o ?p))
   (over all (at ?v ?l))
   (over all (having ?p ?v))
   (at start (>= (battery-level ?v)
             (* (battery-use ?p) 60))))
 :effect (and (at end (sampled ?t ?v))
   (at start (decrease (battery-level ?v)
             (* (battery-use ?p) 60)))) )
```

Fig. 3. The Sample action in PDDL.

the user wants to observe and by selecting payloads the user wants to use for obtaining data. AUVs that are connected to NEPTUS are considered as operational and could be used for a mission. NEPTUS then encodes the information about vehicles and tasks into PDDL, and as a goal sets to acquire all the data specified by the tasks.

The planner decides which vehicle does which task and in which order the tasks are to be performed. Plans follow constraints specified in the action descriptions, i.e., collision avoidance (at most one vehicle can be in one location) and energy constraints (vehicles must not run out of energy before finishing all the tasks) with plans are optimized for total mission time. If the planner is unable to find a plan, the user is notified of plan failure, requiring her/him to iteratively relax constraints. An illustrative example follows:

Example:: An operator needs to plan two AUVs (#'s 1 & 2). #1 has multi-beam, #2 has the downward looking camera, and each of the vehicles has a side-scan sonar. Then, the operator specifies three tasks; two scope out areas of interest, where one has to be surveyed by side-scan and the second by multibeam, while the third refers to a location of another phenomenon that has to be sampled by camera. The planner then assigns the multibeam related task to be performed by #1 and the camera task to be performed by #2. The task where side-scan sonar is required might be performed by both AUVs. The planner decides to allocate it to #2 since it takes less time than #1. However at plan time, it is determined that #2 does not have enough energy; the task therefore gets assigned to #1. When neither of the AUVs has enough energy, the planner does not return any plan, since this last task cannot be allocated.

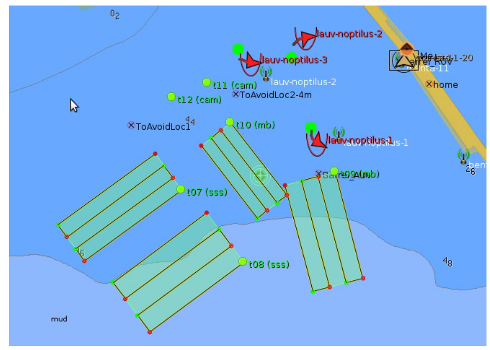
V. Experimental Evaluation

For system evaluation we used a mine-hunting scenario with multiple AUVs with different payloads. Typically, AUVs for such a scenario are equipped with side-scan sonars to image the sea floor, an acoustic altimeter and a high-resolution camera. After side-scan images are inspected by an operator, a set of objects of interest or *contacts* and their locations are determined and the AUVs are dispatched to identify those objects with high resolution cameras to be ground-truthed.

For sea-floor surveys, AUVs typically need to travel while maintaining a constant altitude over ground. This is primarily because for each sonar range and frequency configuration, there is an optimal distance at which the seafloor can be sampled to derive the appropriate resolution of the bathymetry. As a result, planned trajectories must be in accordance to the selected side-scan sonar configurations. Mine-hunting operations are also usually executed with a single AUV or using homogeneous vehicles as plans created manually by operators need to be adapted to each individual vehicle hardware configuration, periodicity of surfacing, side-scan sonar configurations, planned depths, etc.

Conceptually, the use of heterogeneous vehicles for such an application reduces the overall operation time, because some of the vehicles may be better suited for

(a) The deployment area, Leixões harbor, Porto.



(b) Detail of depots and objectives from Fig. 4(a) defined for phase one of the experiment.

Fig. 4. Experiment location and transit lines of the vehicles.

surveying large areas while others may have higher resolution sensors and/or navigation in order to take high-resolution images of detected contacts. However this results in a high cognitive burden on the operator. Moreover, manual planning for AUVs also requires a number of safety procedures that can add to operator overload when using multiple heterogeneous vehicles. Such a scenario is therefore ripe for our evaluation using automated planning.

A. Field deployment

We used 3 LAUV's (Noptilus-1, -2 and -3) in our experiments. All vehicles have different payload configurations but similar navigation accuracy using the same Inertial Navigation System (INS). Noptilus-1 and Noptilus-3 carry lower-resolution Imagenex side-scan sonars and Imagenex DeltaT Multibeam sensors. Noptilus-3 and Noptilus-1 both carry underwater high-resolution cameras and Noptilus-2 carries only an Edgetech side-scan sonar. Noptilus-1 carries an RBR CTD (conductivity/temperature/depth) probe while other vehicles carry a sound velocity sensor, used primarily for correcting sonar measurements.

The deployment area was inside the Leixões harbor in Porto (Fig. 4(a)). The base station was located on top of a pier in the harbor with easy access to the water. The vehicles were deployed from near the base station and tele-operated to the operational area shown in the figure. Inside the operational area, the vehicles were placed in known and safe positions at the surface, which we call *depots*, where communications with the base station was viable. The experiment was then divided in two phases. In the first phase the vehicles were used to survey the

Vehicle	Action	Average Time Difference	Standard deviation
Noptilus-1	move	47,80	49,11
	survey	23,15	23,26
	sample	1,33	0,58
	communicate	0,16	0,17
Noptilus-2	move	39,57	35,66
	survey	107,88	141,10
	sample	N/A	N/A
	communicate	0,25	0,07
Noptilus-3	move	59,90	57,05
	survey	24	0,00
	sample	9,57	13,64
	communicate	0,11	0,16

TABLE I

DIFFERENCE BETWEEN PLANNED AND EXECUTION TIME FOR THE THREE AUVS (N/A IMPLIES A GIVEN ACTION WAS NOT IN A PLAN).

operational area while in the second, to identify and sample in points of interest in the data acquired in the earlier phase [1].

In phase one, the operator defined a set of areas of interest to be surveyed using side-scan or multibeam sonar sensors. Fig. 4(b) is a `NEPTUS` console view taken during the experiment after the operator specified the survey area. In order to task the vehicles, the operator requests `NEPTUS` to generate the plans for the vehicles which will result in translating the current world state to `PDDL` and generating a set of solutions using **LPG-TD**. The best solution is selected automatically based on overall execution time. As a result a set of scripted plans is generated and available for the operator to visualize and simulate. The operator can visually verify the plan and change the objectives and regenerate a solution accordingly if needed.

When the operator has validated the plan, s/he sends the plans to the respective vehicles and order its execution. After the vehicles perform this initial survey, all side-scan data is downloaded to the base station and inspected by the operator in determining contacts in the sampled regions.

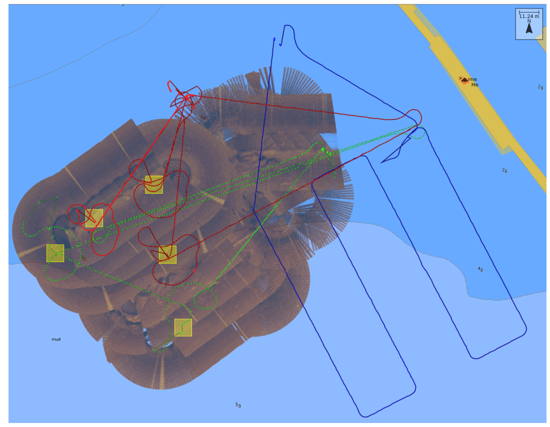
In phase two, a new set of objectives is then defined by the operator in order to get a closer view of these potential contacts. For some contacts it is likely that the operator not only requests a camera inspection but also CTD sampling in order to augment identification. The deployment ends when all vehicles return to their respective depots after completing objectives.

B. Experimental Data and Analysis

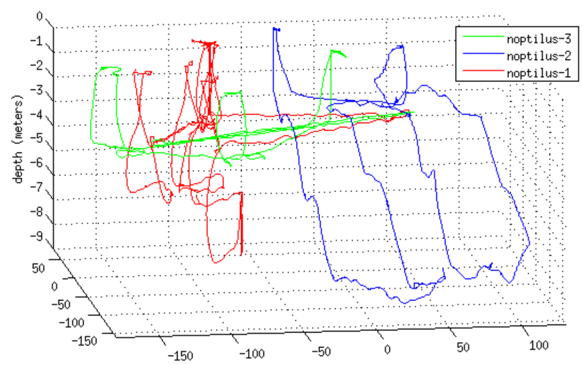
Fig. 5(a) shows a visualization of side-scan data (in brown) that was acquired on the first phase of the experiment showing complete coverage of the survey area. Moreover, the contacts identified by the operator at the end of this phase are indicated with yellow markers. In the second phase of the experiment, these contacts were visited by the vehicles and the tracks performed by the vehicles in phase two are overlayed within `NEPTUS`.

The behavior of the vehicles mapped well into the specification of the operator; however the action durations had discrepancies compared to the predicted

[1]`https://www.youtube.com/watch?v=qWHXRek8_so`



(a) Vehicle trajectory in the second phase of the experiment, overlayed on top of side-scan data (brown) and identified contacts (yellow).



(b) 3D view of the vehicle trajectories for the second phase of the experiment with trajectories from three AUVs.

Fig. 5. Results from phase two of the experiment showing side-scan and trajectories of the AUVs.

plan times. Table I shows the average difference between predicted and actual times for executing different actions. The results show that executing *communicate* and *sample* actions is more accurate than *survey* and *move*. This occurs because *communicate* and *sample* in particular, generate timed actions while the other actions generate a behavior that requires the vehicle to move between different locations whose completion time depends on environment conditions such as water currents, sea floor roughness and time to achieve depth. The roughness of the terrain can be seen in Fig. 5(b) where the trajectories of the vehicles are plotted in 3D. To give an example, the vehicle takes longer to reach the depth required to take a camera image if the location is in a deeper point of the operational area and this depth is not modeled a priori in the planner.

For movement actions, `DUNE` also appended a surfacing behavior. This was added in order to re-acquire GPS positions and therefore improve the quality of localization for upcoming samples and surveys. This is important, since it may require the vehicle to stay underwater for a substantial period of time with consequent localization errors. However this was not reflected in the planning domain model. Since the generated plans do not require time coordination of the vehicles, this did not impact the end result from the operator perspective.

C. Discussion

Incorporating domain-independent planning, i.e., a `PDDL` domain model and the **LPG-TD** planner, into `NEPTUS` does not introduce any serious engineering challenges. Our domain model scales well and **LPG-TD** solves problems with 20 tasks in at most a few seconds. However, the main drawback of our approach is that durations of the actions must be determined a priori, which as shown leads into discrepancies between planned and execution time. Although this is not a major issue for scenarios like ours, more complex applications requiring synchronisation of multiple vehicles might be considerably impacted by such discrepancies accumulating delays. We believe that considering optimistic, realistic and pessimistic estimations of action durations, and then evaluating corresponding plans might help to reduce discrepancies between plan and execution time.

VI. Discussion & Future Work

While the field experimentation validated our preliminary approach for real-world multi-vehicle applications, scenarios that require time coordination among vehicles, the `DUNE` executive lacks in-situ plan adaptation. One trivial approach to solve this problem is to use conservative plans with padded time between actions and an executive that supports timing of actions by advancing to the next action only when its start time has arrived. Such a strategy has indeed been tested with the Networked Vehicle Language (**NVL**) [22].

Moreover, if we consider that the vehicle is connected with other systems only during *communicate* actions, a more advanced executive may deterministically adjust the plan between *communicate* actions. Such plan adjustments may include throttling vehicle speed, changing the traveled path or stopping at the surface, as long as the *communicate* actions, where the vehicles synchronize with other parts of the network, is correctly timed.

Limitations in underwater communication also makes it crucial that part of the temporal execution of the plan for each vehicle is managed locally ensuring local adaptability of vehicle behavior. These problems are precisely where the applicability of a temporal constrained-based planning/execution framework like **T-REX** [9], [10], [11] can be tackled in the near future. While **T-REX** has already been integrated on our LAUVs, coupling multiple **T-REX** enabled vehicles to a shore-based mixed-initiative system as in this paper, is future work. A likely challenge to address in that eventuality, will be the interaction of local and autonomous decision-making with the plan for the ensemble of robots.

In conclusion, we described our work in mixed-initiative control, with a shore-based automated planner synthesizing goals to command multiple AUVs. Abstraction in control is an important objective of this work primarily as a means to control multiple heterogeneous vehicles for distributed problem solving in real-world environments. Finally, we have demonstrated this work by coalescing and abstracting control functions with *one* operator using a well established command/control methodology used in recent field experiments.

Acknowledgements: Rajan and Py are supported by ONR Grant # N00014-14-1-0536 on 'Integrated Autonomy for Long Duration Operations'.

References

[1] D. Yoerger, M. Jakuba, A. Bradley, and B. Bingham, "Techniques for Deep Sea Near Bottom Survey Using an Autonomous Underwater Vehicle," *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 41–54, 2007.

[2] J. P. Ryan, S. Johnson, A. Sherman, K. Rajan, F. Py, H. Thomas, J. Harvey, L. Bird, J. Paduan, and R. Vrijenhoek, "Mobile autonomous process sampling within coastal ocean observing systems," *Limnology & Oceanograhy: Methods*, vol. 8, pp. 394–402, 2010.

[3] J. Gottlieb, R. Graham, T. Maughan, F. Py, G. Elkaim, and K. Rajan, "An Experimental Momentum-based Front Detection for Autonomous Underwater Vehicles," in *ICRA*, 2012.

[4] M. Faria, J. Pinto, F. Py, J. Fortuna, H. Dias, R. M. F. Leira, T. A. Johansen, J. B. Sousa, and K. Rajan, "Coordinating UAVs and AUVs for Oceanographic Field Experiments: Challenges and Lessons Learned," in *ICRA*, 2014.

[5] J. Gonzalez and et.al, "AUV Based Multi-vehicle Collaboration: Salinity Studies in Mar Menor Coastal Lagoon," in *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV)*, Porto, Portugal, May 2012.

[6] J. Pinto, P. Calado, J. Braga, P. Dias, R. Martins, E. Marques, and J. Sousa, "Implementation of a control architecture for networked vehicle systems," in *Proc. IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, 2012.

[7] A. Sousa, L. Madureira, J. Coelho, J. Pinto, J. Pereira, J. Sousa, and P. Dias, "LAUV: The man-portable autonomous underwater vehicle," in *Navigation, Guidance and Control of Underwater Vehicles*, vol. 3, no. 1, 2012, pp. 268–274.

[8] J. Bresina, A. Jonsson, P. Morris, and K. Rajan, "Activity Planning for the Mars Exploration Rovers," in *ICAPS*, 2005.

[9] F. Py, K. Rajan, and C. McGann, "A Systematic Agent Framework for Situated Autonomous Systems," in *AAMAS*, 2010.

[10] K. Rajan and F. Py, "T-REX: Partitioned Inference for AUV Mission Control," in *Further Advances in Unmanned Marine Vehicles*, G. N. Roberts and R. Sutton, Eds. The Institution of Engineering and Technology (IET), August 2012.

[11] K. Rajan, F. Py, and J. Berreiro, "Towards Deliberative Control in Marine Robotics," in *Autonomy in Marine Robots*, M. Seto, Ed. Springer Verlag, 2012.

[12] M. Cashmore, M. Fox, T. Larkworthy, D. Long, and D. Magazzeni, "Auv mission control via temporal planning," in *ICRA*, 2014, pp. 6535–6541.

[13] J. Almeida, C. Silvestre, and A. Pascoal, "Cooperative control of multiple surface vessels in the presence of ocean currents and parametric model uncertainty," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 14, pp. 1549–1565, 2010.

[14] O. Schofield, J. Kohut, D. Aragon, L. Creed, J. Graver, C. Haldeman, J. Kerfoot, H. Roarty, C. Jones, D. Webb, and S. Glenn, "Slocum Gliders: Robust and ready," *J. Field Robotics*, vol. 24, pp. 473–485, 2007.

[15] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa, "The LSTS toolchain for networked vehicle systems," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–9.

[16] M. Ghallab and D. Nau and P. Traverso, *Automated Planning Theory and Practice*. Elsevier Science, 2004.

[17] M. Fox and D. Long, "PDDL2.1: an extension to PDDL for expressing temporal planning domains," *J. Artif. Intell. Res. (JAIR)*, vol. 20, pp. 61–124, 2003.

[18] A. Gerevini, A. Saetti, and I. Serina, "Planning through stochastic local search and temporal action graphs in LPG," *J. Artif. Intell. Res. (JAIR)*, vol. 20, pp. 239–290, 2003.

[19] J. Benton, A. J. Coles, and A. Coles, "Temporal planning with preferences and time-dependent continuous costs," in *ICAPS*, 2012.

[20] J. Frank and A. Jónsson, "Constraint-based Attribute and Interval Planning," *Constraints*, vol. 8, no. 4, pp. 339–364, oct 2003.

[21] M. M. S. Shah, L. Chrpa, D. E. Kitchin, T. L. McCluskey, and M. Vallati, "Exploring knowledge engineering strategies in designing and modelling a road traffic accident management domain," in *IJCAI*, 2013.

[22] E. R. B. Marques, M. A. Ribeiro, J. Pinto, J. Sousa, and F. Martins, "Towards programmable coordination of unmanned vehicle networks," in *Navigation, Guidance and Control of Underwater Vehicles*, 2015.