Talking Autonomous Vehicles: Automatic AUV Mission Analysis in Natural Language

Helen Hastie, Xingkun Liu, Yvan Petillot
Heriot-Watt University,
Riccarton, Edinburgh EH14 4AS, UK
Email: h.hastie, x.liu, y.r.petillot@hw.ac.uk

Pedro Patron
SeeByte
30 Queensferry Road, Edinburgh EH4 2HS, UK
Email: pedro.patron@seebyte.com

***Abstract*—As AUVs are enabled with greater levels of autonomy, there is the need for them to clearly explain their actions and reasoning, maintaining a level of situation awareness and operator trust. Here, we describe the REGIME system that automatically generates natural language updates in real-time as well as post-mission reports. Specifically, the system takes time-series sensor data, mission logs, together with mission plans as its input, and generates descriptions of the missions in natural language. These natural language updates can be used in isolation or as an add-on to existing interfaces such as SeeByte's SeeTrack common operator interface. A usability study reflects the high situation awareness induced by the system, as well as interest in future use.**

I. Introduction

Autonomous systems, by their very nature, act independently making key decisions on which actions to take. They are able to adapt, modifying preset plans and behaving in a less observably deterministic fashion [1]. This behaviour is key to their success and recent wide-spread adoption in a variety of domains and applications (e.g. air vehicles for farmers, underwater pipeline monitoring). However, there can be a lack of clarity in the reasoning behind system actions due to the absence of interactive interfaces between operators and autonomous systems, thus leading to a reduced level of trust. This is particularly prevalent in the underwater domain where one does not have eyes-on the vehicle. Here, we describe one of the first of such interfaces between Autonomous Underwater Vehicles (AUVs) and an operator referred to hence forth as the REGIME system (REport GeneratIon from Metadata).

A. Background

Previous work has looked at enabling the mission plan to be more scrutable and less opaque [2]. However, this work looks at explaining only the plans whereas our work combines the plans with logs of real missions in order to generate accurate reports of what actually happened. Other previous attempts have been made to build natural language interfaces to bring mission states and purposes closer to human operators. However, these are only in simulated environments for pre-mission verifications [1]. The REGIME system described here can process real world missions, with noise-prone data, due to imperfect sensors and irregular trajectories, partially caused by noisy environmental factors (e.g. varying water current, heading and seabed shape). In addition, we are the first to attempt to do this during a mission with only partial information available at any one time.

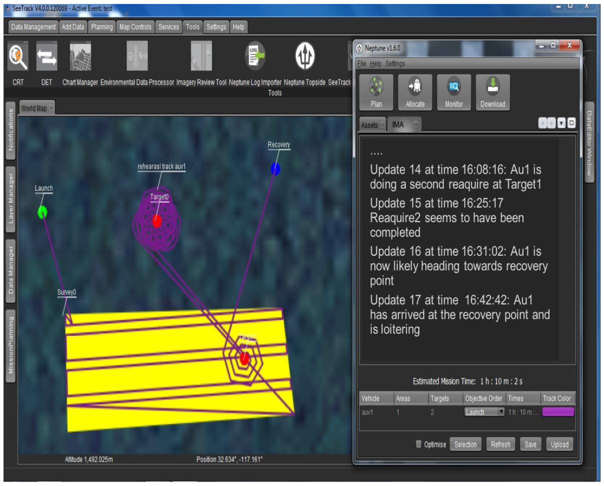

Fig. 1. The SeeTrack interface ©SeeByte with the mission analysis on the right hand side

II. System Architecture

The prototype system described here is able to provide accurate reporting in-real time for In-Mission Analysis (IMA) [3] as well as Post-Mission Analysis (PMA) [4] on both simulated and real data. Both types of generated reports are displayed on SeeByte's SeeTrack common operator interface (see Figure 1). It is designed to generate reports with regards missions running SeeByte's autonomous behaviour software called Neptune, which is platform agnostic and runs on various AUVs such as REMUS and IVER. As illustrated in Figure 2, the report generation module takes three types of data sources as its input: 1) the mission plan; 2) time-series sensor data (i.e. navigation logs); and 3) event/error logs. The PMA system accesses the ROS (Robot Operating System) logs downloaded from the vehicle and extracts the navigation file in the form of comma seperated values document (csv) with information e.g. latitude/longitude/heading/depth. The PMA also takes as input the mission plan and the event/fault logs from the vehicle. The IMA uses similar data in terms of the mission plan and navigation information but takes this directly from the vehicle in real-time via an underwater acoustic communication

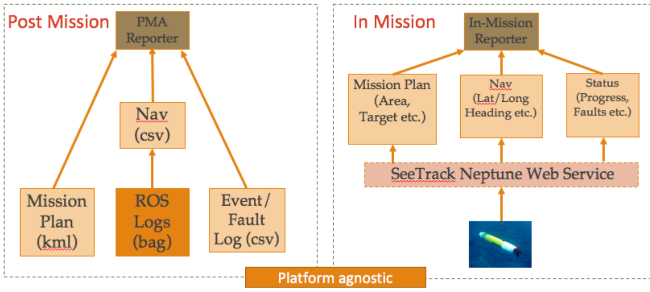

Fig. 2. System architecture comparing Post-mission Analysis- PMA (left) and In-mission Analysis- IMA (right) ©Heriot-Watt

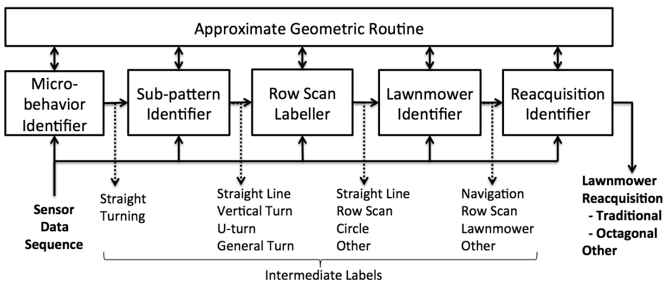

Fig. 3. The Event Identifier: a hierarchical classifier for event identification from real navigation sensor data

link using the Neptune Web Service API. Vehicle status, e.g. faults/warnings, are also obtained in this way.

A. Event Detection

Given a data stream, a sequence of events of interest is extracted by the Event Identifier. This sequence of events is then aligned to the sequence of *planned* events taken from the mission plan. The system focuses on two major event types: wide area surveys (lawnmowers) and close inspection of objects of interest (reacquisitions) [5] for Mine Counter Measure (MCM) missions, where the vehicle is surveying an area for mines for disposal.

For the PMA, this is done using dynamic programming, specifically the Smith-Waterman algorithm. In order to identify events of interest in noise-prone real sensor data, we construct a rule-based hierarchical classifier as shown in Figure 3. To relieve the error accumulation problem, each layer is designed to be robust to imperfect labels from the previous layer. In addition, there is a set of approximate geometric functions specifically designed for this problem to confer robustness to data noise (such as unevenness in row scans or coordinate jumps during navigations)[4] . We refer here to the Event Identifier for the PMA. The in-mission module uses a similar pipeline as that described in Figure 3 but without access to the dynamic programming as the event detection is happening in real time.

The Event Identifier was evaluated on 243 real world AUV missions and 27 simulated data whereby the true sequence of events was transcribed by hand and this sequence was

TABLE I
COVERAGE OF THE DATA EVENT IDENTIFIER FOR POST MISSION ANALYSIS

	REMUS VIP (215)	Neptune (28)	Neptune Sim (20)	Neptune Sim w Excl Zones (27)
Event	79.4%	100%	98.3%	94.6%
Mission	78.6%	100%	95.0%	84.0%

compared to the output of the Event Identifier. Coverage of event detection is given in Table I and calculated as follows:

$$Event\ Detection\ Accuracy = 1 - \frac{(S + D + I)}{N}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions of events and N is the total number of events in the transcription. Mission accuracy is the proportion of missions with 100% of all events correctly identified.

The coverage of the rule based system is presented in Table I in terms of for four sets of data. The first set contains 215 real world missions conducted by the Heriot-Watt Oceans Systems Lab using the REMUS VIP software. The second set contains 28 missions using SeeTrack's Neptune operating software running on both IVER and REMUS vehicles. Note that the rule-based system was tuned on this dataset and therefore gets 100% coverage.

In order to test the true capability, the system was run on simulated data, which introduced some noise. Specifically, 27 single vehicle missions were simulated using the Neptune Simulator and Neptune Topside with settings to create a variety of scenarios. Noise conditions were applied to some simulated missions, such as changing the water current speed and direction, changing the seabed shape (ripple or complex), adding dynamic targets for reacquisition, changing shapes and positions of the survey areas and adding exclusion zones. In addition, a number of simulations were run with event trigger scenarios e.g. dynamically changing the water current direction 10 minutes after launching the mission. The vehicle would try to adapt its direction based on the water current direction after such an event is triggered. The third set of data in Table I reports the coverage of the system on these simulated data without exclusion zones where coverage reaches 98.3% and 95% for Event and Mission respectively. The fourth set includes exclusion zones, where there is a degradation of performance when these exclusions zones are put in place. Given that the system was not designed to cover missions that included exclusion zones, this level of coverage is encouraging (94.6% and 84%). Table I reports on event detection for post-mission analysis but we do not predict much of a degradation for in-mission reporting on the assumption of continuous comms.

B. Generation of Natural Language Reports

The set of recognised events, along with other items of interest are encapsulated in the form of formal semantics, which is then sent to the Natural Language Generation (NLG)

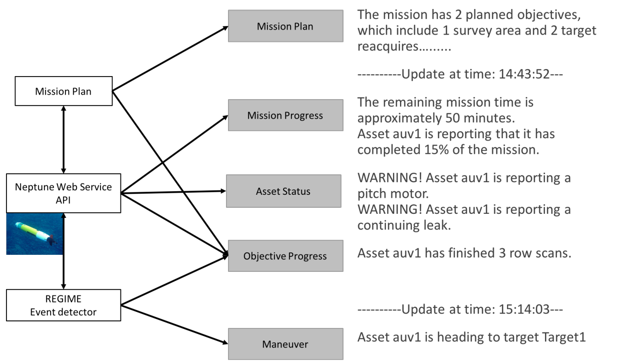

Fig. 4. System architecture for in-mission reporting with example outputs ©Heriot-Watt

Report Example	
Semantics	**Report Text**
MSN04 InformTask(lawnmower=1, reacquire=2) InformTask(task=reacquire, target=[32.6354; -117.1643]) InformTask(task=reacquire, target=[32.6334667; -117.162233]) InformSuccess(task=lawnmower) InformUnfound(task=reacquire) InformNewEvent(task=reacquire, target=[32.6262; -117.1641]) InformSuccess(task=reacquire)	This mission includes one lawnmower scan followed by two reacquisitions. The first target to reacquire is [32.6354;-117.1643]. The second target to reacquire is [32.6334667; -117.162233]. The lawnmower scan was finished successfully. A reacquisition was completed but at location [32.6262;-117.1641]. A second reacquisition was successfully completed.

Fig. 5. PMA for mission given in Figure 1 with the semantic representation on the left hand side and the example output on the right

component. Figure 5 gives an example of the formal semantics on the left-hand-side and the realised PMA on the right. For the post-mission reporting, the natural language generation system is broken down into a number of individual reporter modules that use rule-based generation. These are:

- **Mission Plan Reporter** debriefs on the plan of the mission;
- **Plan Success Reporter** debriefs on the planned events successfully executed by the AUV;
- **Out-of-plan Event Reporter** reports on events that are outside the original plan e.g. dynamic object reacquisition;
- **Abort Reporter** seeks the reason for the abort, locates and reports the aborted mission objective;
- **Mission Log Reporter** reports important messages in the mission log, such as automatic parameter adjustments by the AUV, the thruster controller errors during the mission, etc.;
- **Data Error Reporter** reports potential errors identified in the sensor data such as coordinate jumps (e.g. due to noise in the navigation sensors).

The in-mission analysis system also contains the **Mission Plan reporter** module, but includes the following modules as illustrated in Figure 4 with example output text:

- **Mission Progress Reporter** debriefs on status with relation to the mission objectives;
- **Asset Status Reporter** debriefs on any warnings coming from the asset;
- **Objective Progress Reporter** debriefs on progress status for the current objective (e.g. doing a survey or a reaquisition);
- **Maneuver Reporter** debriefs on status if the asset is maneuvering e.g. towards the starting location of a new objective or recovery point.

III. Personalisation

For the system design, we deployed user-centred design whereby we did an initial data collection to understand the information needs of users and how these information needs differed depending on the role of the user. Participants included GCS (Ground Control Station) Operators and Technical managers, Operations Command, Maintenance crew and Processing, Exploitation and Dissemination personnel namely environmental scientists. For the study, the users watched three videos and were asked to select the type of information they would like to see (e.g. mission type, launch time, mission success). A report was then automatically generated, which they could then edit for wording and style. We also performed interviews with various potential end-users, to understand their information needs further and determine what information challenges they face in their role with regards to AUV missions, situation awareness and training. From these studies, it was determined that users do have different information needs. Examples of these differences are given below:

- **Operations Manager** want to know all possible information but in a succinct manner;
- **Maintenance Crew** want to know only the information whereby the mission went wrong and that which will inform training of personnel;
- **Processing Exploitation Dissemination** want to know information only pertaining to the collection of valid data.

To this end, a personalisation interface (see Figure 7) was implemented, which allows for information customisation through the SeeTrack interface.

IV. Usability study

A usability study was conducted with 15 users in the field of AUVs. These subjects were software engineers and operators with varying experiences and computer programming expertise. The study was between-subjects design with two conditions where one group was shown a sequence of utterances generated by the system (‘language only condition’) and the other group was given both language and the visual path and estimated vehicle position (‘multimodal condition’) as in Figure 1. At the end of the mission, subjects were asked questions to garner an idea of their overall mental model of the system in terms of understanding what the system is doing and how it is working as well as general usability.

The questions from the post-questionnaire are listed here:

- Q1: In general, I trusted the AUV was doing what it should be doing
- Q2: I have an overall understanding of how the AUV was working

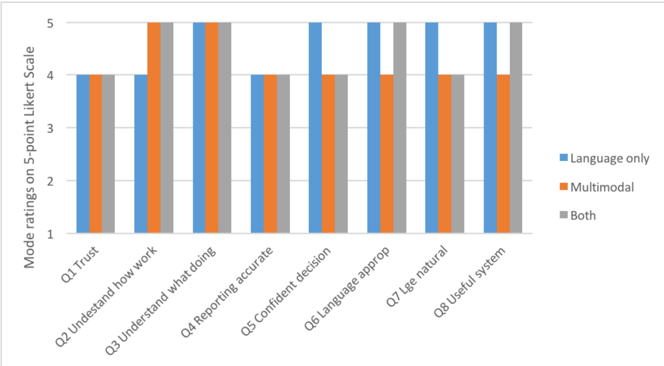

Fig. 6. Mean responses for post-questionnaire on usability

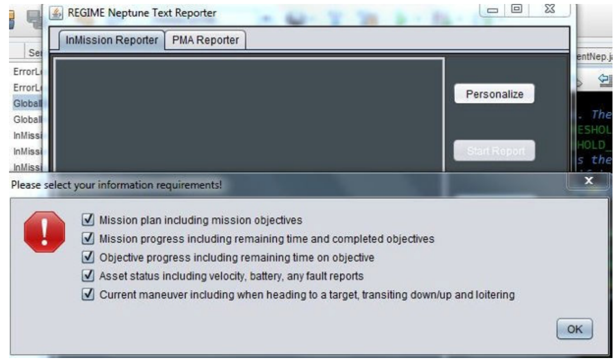

Fig. 7. Interface for personalising information reported

- Q3: have an overall understanding of why it was doing what it was doing
- Q4: I have confidence that the reporting was an accurate reflection of what the system was doing
- Q5: I would feel confident making a decision whether to abort the mission or not based on this reporting
- Q6: The language was appropriate
- Q7: The language used in the reporting was natural
- Q8: I would find it useful to have this type of reporting during the mission

Overall the system received high usability ratings (see Figure 6) where *mode=5* on a 5 point Likert scale for understanding how the system was working and what it was doing on the mission, thus reflecting a high situation awareness induced by the natural language. We also observed a *mode=5* for language appropriateness, reflecting the effective design of the natural language generation system as well as a *mode=5* for whether subjects would use the system in the future. With regards to the two conditions, subjects expressed the same level of trust in the system across conditions of *mode=4*, however, for the multimodal system there seemed to be a more accurate mental model of what the system was doing (*mode=5* compared to *mode=4*), reflecting the need for both visual and language output for maximum situation awareness[1].

Qualitative feedback included statements such as “In general, these types of messages are great and informative.” and “The reporting was good in that it reported the most important information, at a good pace”. Some areas for improvement included the users wanting to know the source of the information.

V. Discussions and Future work

This paper describes how natural language reports are generated for real and simulated AUV missions using metadata to infer the events in the mission, combined with mission plans and system self-reported errors. A usability study reflects the high situation awareness induced by the system as well as high interest in future use.

[1]There are too few data points to do statistical analysis with high enough effect-size

REGIME here deploys a sophisticated method for event detection that avoids error accumulation but one that is rule-based. Machine learning techniques have been proposed in previous works for AUV behaviour identification based on simulated missions [6], as well as for NLG on time-series data [7]. However, besides the labour-cost of data collection itself, the expertise required for data annotators and the confidential issues related to real AUV missions are also main barriers of practically applying large-scale data-driven methods to real world missions [8]. Investigating highly data-efficient learning algorithms for pattern recognition in AUV trajectories and for NLG with limited data will be the focus of our future research. Future work also includes, report generation from more complex missions involving multiple collaborative agents on multiple platforms. These mission types will lead to more challenging NLG problems requiring, for example, aggregation and referring expression generation.

It is understood that we are reporting in a comms-limited environment, however, the experimentation reported here assumes comms are present. Whilst, improving comms is out of the scope of this work, future work will look to work with the latest comms technology and adapt to missing and incomplete data as a result of the limited comms.

An information level customisation interface was developed as described here and shown in Figure 7. Future work would look to develop adaptive personalisation where, through online machine learning, the system would learn and understand the information needs of the user, also recognising cognitive overload and instigating mitigations strategies to optimise situation awareness whilst still managing cognitive load. Finally, we would like to develop a truly interactive system, whereby the user can ask questions regarding missions and interrogate the data either through text, chat or voice. Such an interactive system would provide a medium through which the system could learn user’s preferences and information needs and adapt accordingly through interaction rather than one-way reporting.

VI. Acknowledgements

This research was funded by Dstl (0913-C2-Ph2-009). We acknowledge Dr. Zhuoran Wang and Thales for their contribution.

References

[1] N. Johnson, P. Patron, and D. Lane, "The importance of trust between operator and AUV: Crossing the human/computer language barrier," in *Proceedings of OCEANS*, 2007.

[2] N. Tintarev and R. Kutlak, "SAsSy- making decisions transparent with argumentation and natural language generation," in *Proceedings of IUI Workshop*, 2014.

[3] H.Hastie, X. Liu, and P. Patron, "A demonstration of multimodal debrief generation for AUVs, post-mission and in-mission," in *Proceedings of ICMI*, 2016.

[4] Z. Wang and H. Hastie, "A prototype for AUV post-mission debrief generation from metadata," in *Proceedings of AAMAS*, 2015.

[5] P. Patron, Y. Pailhas, J. Cartwright, F. Maurelli, Y. Petillot, J. Sawas, and N. Valeyrie, "Fully integrated multi-vehicles mine countermeasure missions," in *Proceedings of UAM*, 2011.

[6] M. M. Novitzky, C. Pippin, T. R. Collins, T. R. Balch, and M. E. West, "AUV behavior recognition using behavior histograms, HMMs, and CRFs," *Robotica*, vol. 32, no. 2, pp. 291–304, 2014.

[7] D. Gkatzia, H. Hastie, and O. Lemon, "Comparing multi-label classification with reinforcement learning for summarisation of time-series data," in *Proceedings of ACL*, 2014.

[8] H. Hastie, "Challenges of military applications for data-to-text generation," in *Proceedings of the 1st International workshop on Data-to-text Generation (D2T)*, Edinburgh, UK, 2015.