

Identifying Error in AUV Communication

Joseph Coleman

Center for Intelligent Systems Research, University of Idaho, Moscow, ID 83844-1024

Kaylani Merrill

Center for Intelligent Systems Research, University of Idaho, Moscow, ID 83844-1024

Michael O'Rourke

Center for Intelligent Systems Research, University of Idaho, Moscow, ID 83844-1024

Andrew G. Rajala

Center for Intelligent Systems Research, University of Idaho, Moscow, ID 83844-1024

Dean B. Edwards

Center for Intelligent Systems Research, University of Idaho, Moscow, ID 83844-1024

Abstract – Mine Countermeasures (MCM) involving Autonomous Underwater Vehicles (AUVs) are especially susceptible to error, given the constraints on underwater acoustic communication and the inconstancy of the underwater communication channel. Little work has been done to systematize error identification and response in AUV communication. We introduce a systematic approach involving Design Failure Mode and Effects Analysis (DFMEA) that is adapted to the complex character of communication between autonomous agents.

I. INTRODUCTION

Communication is an essential component of cooperation among AUVs. For AUVs to coordinate their actions, they must share information and make requests. When AUVs are on the surface, they can use RF (radio frequency), but they are otherwise limited to acoustic communication. The underwater channel is notoriously bad for communication, with propagation effects such as ray bending and multipath adversely affecting communication. Data rates are also restricted because acoustic energy is absorbed by water as the frequency increases [1]. The current method of dealing with error in underwater communication is to build redundancy into the code and large amounts of error correction processing into the receiving end. An alternative approach has been to borrow from natural language semantics and pragmatics in designing flexible AUV languages that reduce reliance on processing resources, thereby minimizing error rates. It is clear, however, that communication error is a major thorn in the side of cooperating AUVs.

Cooperation among AUVs is needed to accomplish increasingly difficult tasks. AUVs have contributed to the US Navy's underwater Mine Countermeasures (MCM) by finding, classifying, and neutralizing underwater mines; they have, however, been limited to single vehicles acting independently. The US Navy is moving towards large area MCM with complete coverage requirements of 30 km by 30 km in a week. Given the current AUV coverage rate, deployment of multiple vehicle formations will be required. Since the ocean is a dynamic, unpredictable environment where all relevant events cannot be anticipated, AUVs in the formation will need to handle problems cooperatively during the mission; otherwise, time will be wasted in covering missed areas.

Replacing a lost vehicle [2], dealing with lost communication [3], and acquiring mine location targets [4] are aspects of cooperative behavior we have investigated. As collaborative behavior has grown more sophisticated, the

potential sources of error have also grown. Since we are dealing with autonomous agents, the errors go beyond simply corruption of the signals transmitted from one vehicle to another. They can involve, for instance, the theory of agency we introduce into our system—if an AUV's beliefs about the environment or the other vehicles are incorrect, it may transmit an incorrect message or falsely interpret an incoming message. These errors can exert their influence at any time during the run, producing different effects depending on various aspects of context, including what has happened during that run. Given the complexity of the AUV communication system, we should approach error classification systematically so that we can better control its identification and our response. This approach should encompass the whole communication system, from the sender's initial message planning through transmission of the signals to the receiver's interpretation. At present, we know of no such approach in the literature.

Our approach requires thinking of system error as a type of system failure—error is induced into the system when an element of the system fails to do its part, implying that system error and system failure are closely correlated notions. We exploit this correlation for the purpose of structuring error identification and response. To this end, we introduce Design Failure Mode and Effects Analysis (DFMEA), which is an existing approach to modeling system failure. DFMEA divides the system into functional modes and isolates specific forms of failure in terms of a cause/effect pairs associated with each mode. We have modified the DFMEA for a communication system involving autonomous agents, where the same error can propagate into many different effects.

II. ERROR AND AGENT COMMUNICATION LANGUAGES

A. Introduction

AUVs are autonomous agents capable of complex and flexible behaviors. Their autonomy is made possible by sophisticated software that enables them to perceive, reason about, and adapt to their environment. As such, they are a type of *agent*, understood by AI researchers to be a software program designed to autonomously perform similar tasks. Agent researchers recognize that achieving some tasks requires the collaborative effort of multiple agents, i.e., agent *interoperability*. Agent research has focused on this interoperability requirement with a view to facilitating collaborative execution of tasks carried out by interacting agents. In order

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 2006		2. REPORT TYPE		3. DATES COVERED 00-00-2006 to 00-00-2006	
4. TITLE AND SUBTITLE Identifying Error in AUV Communication				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Idaho,Center for Intelligent Systems Research,Moscow,ID,83844				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 8	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

for a collection of agents to cooperate effectively, they must be able to share knowledge, take advantage of the capabilities of other agents, and exploit mutually available resources. Agent communication languages (ACLs) are designed to be a mechanism for facilitating this type of cooperation between agents.

B. Error and ACLs

ACLs are complex and multi-faceted. We can distinguish between lower level and higher level aspects of ACLs, with syntax and semantics at the lower level and pragmatics at the higher level. One potential complication introduced at the lower level involves open agent system. In closed agent systems, agents are homogeneous, sharing the same software profile. In open agent systems, agents are heterogeneous, creating the potential for translation errors across profiles. Presently, most AUV architectures are homogeneous, but deployment of AUVs in MCM will require formation/fleet communication in addition to intervehicle communication.

At a higher level, rules of interaction must be established for cooperating agents and an ACL must contain a protocol for agent interaction. Greaves, et al. identify this as a necessary pragmatic component of ACLs and use the term *conversation policies* for this protocol [5]. Conversation policy violations are a type of communication error that must be identified and resolved by communicating agents. Protocol violations may be a result of lower level errors in communication. For example, an agent may induce a protocol violation by failing to understand a message or by failing to process an already erroneous message; to further the problem, the protocol violation may not be immediately identifiable but may propagate further error.

C. AUV ACLs

One language representative of the effort to create a common language for cooperating AUVs is COLA [6]. Developers of COLA recognize that high error rates are characteristic of AUV communication, given the unreliability of underwater communication and significant limitations on bandwidth and available resources for message processing. In [6], the developers of COLA note that the possibility of a message containing an error must be considered while designing a language for AUV communication and cooperation. Because communication between multiple AUVs is in the service of cooperative tasks, communicative success translates into mission success. To communicate successfully in an error prone environment, designers must anticipate the prevalent types of error and design AUVs so that they can detect them.

COLA developers specifically rely on syntactic parsing to identify errors. Legal message structures are prescribed and a syntactic parse fails if the message contains errors that violate the prescribed message structure. At this level of error detection, errors affecting message form are identified first. Some errors will not be induced in the form of the message but will be more conceptual; thus, AUVs must recognize defective messages with proper syntax. Given this, more robust error detection must be introduced. Once a message has been determined to have a legal syntactic structure, the AUV must

test each parsed component of the syntax to make sure that the message has an acceptable semantic structure.

COLA implements a semantic interpreter to identify such errors. In COLA, the results of the syntactic parse are sent to the semantic interpreter, where the semantics of the message received are checked against semantic constraints. If the constraints are violated, the semantic interpreter fails and does not process the message. Turner, et. al, offer an example of an error in which a syntactically appropriate message contains a semantic violation which would send an AUV below crush depth. To recognize that this value cannot be used, the receiver must check the message against constraints associated with crush depth and return an error. It is noted that this sort of error could be introduced during transmission, or it could be generated by the sender. Either way, it is clear that a variety of errors will be introduced and thus must be managed. To build such accountability into language design, a systematic view of the levels at which different types of error can occur must be developed.

III. DESIGN FAILURE MODE AND EFFECT ANALYSIS

A. What and Why

As we have argued, communication failure among AUVs will be common, due to acoustic properties of the water, low bandwidth, and complexity of the communication systems. The problems of underwater communication have been well documented, but little has been done to examine how the communication problems combine with the complexities of cooperative behavior to affect the overall goal of complete coverage. We propose using a DFMEA in the communication design process to account for failure and its effect on overall system goals. DFMEA is a tool used to analyze failure in a system and plan a design response to that failure, on the assumption that system failure is the result of regular causal processes that can be identified and avoided. It also gives us a method for documenting failures in the system and the actions taken to deal with them.

DFMEA enables a design team to take a snapshot of the design process for a particular system. They can use this to record what has been done, what is and is not working, and where they must go if they are to achieve their design goals. The tool is applied iteratively at various points in the process to ensure that progress is being made. It is documented in a table with a number of columns that systematically individuate aspects of failure identification, rating, and response. The first step in a DFMEA is for the team to develop a rating system for the occurrence, severity, and detection of failure. The ratings are to remain constant so the first DFMEA can be directly compared to the last.

Those columns associated with failure identification include "Item and Function," "Potential Failure Mode(s)," "Potential Cause(s) of Failure," and "Potential Effect(s) of Failure." In filling out the table, one begins by identifying a part of the designed system (i.e., the "item," with its associated "function") and then distinguishing the different ways or *modes* in which it can fail. These are further distinguished in terms of specific causes and effects that can give rise to these modes, along with associated occurrence and severity ratings.

Those columns associated with failure response include "Current Design Controls," "Recommended Actions," and

“Action Results.” After identifying specific instances of failure, the design team lists the current design controls along with detection ratings. The Risk Assessment Number (RPN) is calculated by multiplying the three rating together. This rating determines which cause/effect failure pairs are most critical. Recommended actions are rated when implemented to determine if they adequately reduce the RPN. The final column records the actions taken and their effects on the ratings. More information is added to the DFMEA table as design proceeds and more information is known [7].

B. Our Approach

Design processes are essentially goal oriented. Application of the DFMEA is driven by a backward-looking desire to avoid past failure and a forward-looking desire to achieve design goals. The tool represents these desires in its “Recommended Actions” column, which serves to record actions designed to balance failure avoidance with goal pursuit.

There are different ways to apply the DFMEA in the course of a design process. We distinguish two: the *mereological* and the *teleological*. The mereological, or part/whole, approach involves applying the DFMEA one part (or stage) of the process at a time. This “one step at a time” approach treats each design stage individually, modifying the item/function and failure modes from iteration to iteration. (For an example of this approach, see [8].) The teleological, or means/end, approach involves creating a DFMEA that is explicitly keyed to the final goal of the design process. Knowledge of the specifications associated with eventual system success can be used to identify failure types that must be avoided along the way. This “ends determine the means” approach treats each design stage relationally as a means to the end product under development, and so the item/function and failure modes remain static across iterations of the DFMEA.

For instance, a mereological approach to communication system design might begin with a DFMEA keyed to the stage of language development, focusing on the limited goal of creating a workable language for information exchange. As the design team moves to the logic and agent theory, the DFMEA would be recast to fit this stage, focusing on different items and modes of failure. A teleological approach, by contrast, would install items and modes of failure determined by the nature of the system to be developed, and these would shape failure identification and response throughout the design process.

We adopt the teleological approach in this paper, for two reasons. First, the nature of the communication required for successful AUV interaction is fairly well-known, at least in broad outline, and our goal is to design a system realizing communication of this sort [9]. Second, much is known about communication failure in the psychological and linguistic literature, and by organizing the DFMEA in this fashion, we can borrow insights from those fields [10]. The failures that most concern us are those that would directly impact eventual system success, and so we use our antecedent knowledge of that success to guide identification and response to failure.

C. The Structure of the DFMEA

As noted above, the DFMEA is a table with columns devoted to aspects of failure identification and failure response. We now detail our treatment of these in turn.

1) Item and Function. The communication system to be designed will support information exchange between AUVs. Following [11] and [12], we model this as signal propagation from a sender through a transmission channel to a receiver. Fig. 1 presents our model. We presume that each item operates as a module, and that we can assess the system for failure in a modular way, looking at each in turn independently of the others.

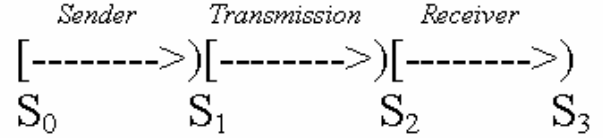


Fig. 1. A model of the AUV communication system used to frame failure mode identification.

The sender is an AUV engaged in generating and transmitting a message, although it may fail to produce one. It is represented as the segment between S_0 and S_1 in Fig. 1. S_0 marks the initiation point of the communication process. We presume that all is well at that point, and indicate that presumption with the square bracket. Our analysis of sender failure will be conducted at S_1 , looking back on the processes in the sender that should function to produce the appropriate message. We indicate this with the round bracket.

AUVs communicate with one another via acoustic modem, and so the transmission channel will typically be water; however, AUVs may also need to surface and communicate with the fleet via radio connection. In Fig. 1, the channel is the segment from S_1 to S_2 . We focus on underwater communication in our DFMEA. Again, we assume that all is well at S_1 and evaluate the success of the communication system from there through S_2 .

The receiver is an AUV tasked with the job of receiving and interpreting the message. We represent the receiver as the segment from S_2 to S_3 , and assume that the signal has arrived at the sensors at S_2 intact and error-free. We assess at S_3 , looking for failures introduced in reception and interpretation.

2) Modes. A failure *mode* is a form or type of system failure. This part of the DFMEA serves to classify broad types of system failure. Failure modes are further analyzed into associated causes and effects in the next two columns. These modes can be individuated in a variety of ways, but here our teleological approach inclines us to identify stable and systemic ways of identifying failure types. Given this approach, the modes should be determined by the character of successful intervehicle communication and remain the same throughout the many iterations that might occur in the design process. The identified modes should reflect a way of examining signal propagation and manipulation from the perspective of overall design success.

If successful, the communication system will support propagation of a signal from sender through transmission channel to receiver. We adopt the point of view of the signal as it propagates through the three functional stages. At each

evaluation point, we can ask whether there is a signal (or interpretation, in the case of the receiver) or not. In addition, we can ask whether there should be a signal/ interpretation or not. Crossing these gives us a 2x2 modal analysis—see Table 1. Three of the four cells can harbor possible failures, the lone

TABLE I 2 x 2 Matrix for Analyzing Failure Modes		
Should there be a signal/interpretation? Is there a signal/interpretation?	Yes	No
Yes	If (a) incorrect, (b) incomplete, or (c) garbled, then: <i>Failure</i> . Else: <i>No Failure</i>	<i>Failure</i> : Signal when there shouldn't be one
No	<i>Failure</i> : No signal when there should be one.	<i>No Failure</i>

exception being the “Is-Not/Should-Not” cell. The “Is/Should-Not” and “Is-Not/Should” cells are home to obvious failures, and the “Is/Should” cell can be problematic as well if the signal/interpretation is not correct, complete, or intact, i.e., if it is not the one that should be there.

From S_1 , we ask if there is a signal and if there should be one, assuming all is well at S_0 . If there is a signal and there should be one, then all is well, unless the signal is incorrect, incomplete, or garbled. By “incorrect” we mean to indicate that the AUV might send the wrong message, due to failure in some aspect of the system. From S_2 , we engage in the same analysis. As we assume that all is well at S_1 , we should expect signal erosion failures at S_2 if we find any failure at all, unless there is noise or an intentional attempt to sabotage the communication transaction. Our analysis at S_3 is slightly different, as we are looking at the endpoint of signal propagation. At this point, our focus is interpretation. Thus, we ask if there is an interpretation and if there should be one, leading to an analysis that is structurally similar to the preceding two.

3) Causes and Effects. Within each mode, there will be a variety of cause/effect pairs that correspond to specific ways in which the item/function in question can fail. We know of failure through its effects. Were part of the system to fail (i.e., were it to stop functioning as it was designed) without discernible effects, either direct or indirect, then we would have no evidence of that failure and no reason to be concerned about it. It would be innocuous and would go unnoticed and untreated. Thus, we first look for effects associated with our failure modes at each of our evaluation points and then look back to the item under examination for causes of those effects.

While we could list cause/effect pairs in no particular order, we choose to provide a conceptual framework for organizing these pairs and our subsequent responses to them. This framework reflects our conception of the AUV communication system as involving *agents* capable of autonomous and flexible responses in unpredictable circumstances. Complex hardware and software support these agents in their communicative efforts, but the nature of these efforts is to some degree *intelligent*. This permits us to engage with the AUVs as purpose-driven actors whose actions are more or less *rational*, i.e., more or less coherent with their communicative goals. Indeed, the AUVs will engage with one another in this fashion, sending messages and interpreting messages in collaborative pursuit of mission goals. Following [13], we call this level of engagement the *intentional* level. Of course, the extent to which this type of engagement is available will vary, depending on the agent theory coded into the AUVs and the language used by them. (For example, the more the language encodes a command-and-control structure, the less flexible and intelligent the participating AUVs will tend to be.)

One way we might respond to failure is by adopting a hardware-first default posture, whereby we begin by examining the hardware for causes when confronted with any non-obvious failure. We believe that this posture will not prove efficient as a way of dealing with a communication system involving complex agents—it would be analogous to performing invasive surgery whenever a person exhibits a health problem. Applying [13] once again, we distinguish two other levels of engagement, viz., the software level and the intentional level. Once again an analogy with humans is appropriate: if there is a behavior problem, we can engage with a person intentionally and address the causes of the problem by reasoning with them; if that fails, we can engage with them psychologically through counseling; if that fails, we can engage with them physically through medical treatment. The degree of invasiveness increases as one moves from the intentional level through the software level to the hardware level.

Thus, we can maximize response efficiency by dealing with the causes of failure first at the intentional level, and then moving through the software level to the hardware level only when our response options are exhausted. When dealing with agents as complex as AUVs, there is no reason to adopt a very aggressive hardware-first default posture; by taking advantage of the three levels of engagement, we minimize response aggressiveness and thereby maximize response efficiency.

In light of this way of looking at the system, we classify cause/effect failure pairs as *intentional*, *software*, or *hardware*, depending on the level at which we locate the cause of the failure. When we evaluate an item from one of the evaluation points, we begin by identifying effects associated with each failure mode. For example, we might find at S_1 that the sender is failing to send a message that should be sent, or perhaps it is sending the wrong message. A failure will be described in terms of its associated mode, and so described, is independent of the three-part distinction used to classify causes.

We begin our search for causes at the intentional level. Here the causes are described in intentional terms, e.g., ‘belief’, ‘desire’, ‘intention’, ‘goal’, ‘plan’, etc. If the cause can be described in this way, the remedy will be straightforward and will not require software or hardware repair. This level of

analysis will be available for sender and receiver but not for the transmission channel. Given the S_1 effects from the previous paragraph, we would investigate whether our sending AUV can be credited with mistaken “beliefs” about the time of its turn to communicate or the physical situation it is in that could be corrected by a simple transmission. (See the pair E_1/C_1 in Fig. 2.)

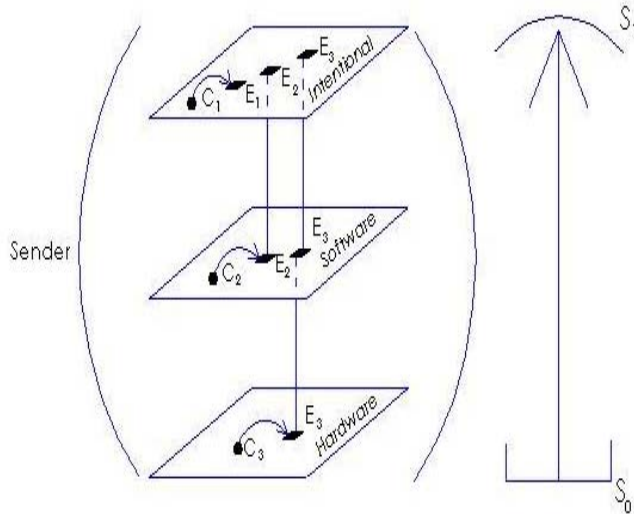


Fig. 2. Model of failure detection in the sender. We identify the effects of failure at the intentional level and then pursue causes for those effects through the three levels. Our investigation begins at the intentional level, which corresponds with our initial engagement with the sender at S_1 .

If we are unable to identify a cause at the intentional level, we move to the software level and examine whether the cause is located in the code or the logic of our agents. As with the intentional level, this level will in general be unavailable when dealing with the transmission channel. Returning to our sending agent, is it failing to transmit because of a bug in its code? Perhaps there is an interaction effect in the various system logics that has interfered with transmission. In general, remedies of causes at this level, such as repairing code, will be more invasive than repairs at the intentional level but less invasive than hardware repair. (See the pair E_2/C_2 in Fig. 2.)

Finally, if we are still unsuccessful in our search for the cause, we move to the hardware level, i.e., the level of the physical systems that implement code and support communication. This level is available for all items. With respect to our sample failures, there may be a problem with the modem, or with the hardware that implements the communication software. As repairs at this level are time-intensive and costly, it is best to avoid them if at all possible; however, there will be occasions when we have no choice. (See the pair C_3/E_3 in Fig. 2.)

We have illustrated the search for causes at S_1 , but the same methods hold true at the later evaluation points, depending on the availability of the levels of analysis. If the search for the cause is repeatedly unsuccessful at any level, then we are left with two options. First, we might bracket the failure and press ahead in the hope that the failure will “work

its way out in the wash.” Alternatively, we might reject the assumption of modularity above and embrace the possibility that the effect really emerges from a combination of causes located in more than one item in the system. Thus, there could be causal influence that “builds” through S_1 and S_2 without resulting in any noticeable failure, only to reach a failure threshold before S_3 . In that case, the DFMEA we have designed should still work to support one in tracing causal influence back through preceding items in the system.

4) Design Responses to Failure. There are three columns for design responses to identified system failures: “Current Design Controls,” “Recommended Actions,” and “Action Results.” The first of these records extant aspects of the system designed to control and protect against failure. Once we identify relevant cause/effect pairs associated with each failure mode, we ask what features of the current design are in place either to mitigate the effects or control the causes. These go in the “Current Design Controls” column. This column represents the starting point for design responses to failure. We may be able to exploit existing controls to mitigate failure, but this will not always be possible, especially early in the design process.

In those cases where current controls are insufficient, pursuit of design goals requires recommending new actions, e.g., modifying the functionality of existing systems or introducing new systems. These recommendations are included in the “Recommended Actions” column, associated with the cause/effect failure pairs they are intended to address. Minimally, the action might be to stand pat, if the failure does not threaten overall system function and it is more cost effective to marginalize it than repair it. In most situations, though, the actions will be more aggressive. These will be distributed across the three causal dimensions. The current software/hardware configuration might enable us to respond intentionally to the failure, modifying the state of the system sufficiently to avoid future failure. For example, a vehicle’s memory might become incorrect during the mission due to a missed communication. An intentional fix would be to have the AUV leader correct the vehicle’s memory during its next broadcast. Here we would use existing controls to immunize the system against future failures of this type. Occasionally, though, we will need to act at the software or hardware levels to address the failure. At those levels, an attempt is made to prevent the cause of the failure by making changes to the vehicle. For example, the logic may be designed incorrectly, causing persistent inaccuracies in memory that take too many formation resources to correct. Failures of this sort would require alterations on the software level. Alternatively, the problem may be rooted in the hardware—a sensor may be malfunctioning or perhaps the AUV lacks a sensor needed to keep the memory current.

The final column is reserved for what results from the recommended actions. The first sub-column in this part of the DFMEA records the action taken in response to the identified failure. The remaining sub-columns record the impact this action is taken to have on the ratings that measure the severity, occurrence potential, detectability, and RPN, or overall impact on the system. To these measures we now turn.

D. Ratings

We have tailored the occurrence, severity, detection, and risk assessment number ratings to the task of underwater MCM with cooperating AUVs. The occurrence rating measures the frequency of failure causes—see Table II. The occurrences are much higher than a normal DFMEA because the same communication error can happen several times throughout the same mission. Based on our work involving the replacement of lost vehicles, 1 in 100 missions was considered very low and 100 times a mission was considered very high.

TABLE II Occurrence Rating		
Score	Rating	Description
1	Very Low	1 in 100 missions
2	Low	1 in 10 missions
3	Medium	Once a mission
4	High	10 times a mission
5	Very High	100 times a mission

The severity rating measures the impact of failure effects on the system. Our severity ratings, shown in Table III, are based on how the effects of failure impact the overall fleet goal. The rating is low if the failure causes the formation to assign a vehicle to a different position, slightly increasing search time. The rating increases if the formation loses the ability to finish the search because there are too many gaps or it has lost too many AUVs. A medium severity rating is assigned to failures that cause the fleet commanders to change the search strategy. A scratched mission earns a high rating—the MCM operation is in support of other military operations and scratching the MCM mission can severely impair those operations. The final and highest rating is assigned to failures that lead to lost ships or loss of life.

TABLE III Severity Rating		
Score	Rating	Description
1	Very Low	Lose time and efficiency
2	Low	Lose ability to search with this formation
3	Medium	Change in strategy
4	High	Scratched mission
5	Very High	Dead sailor, sunken ship

In a typical DFMEA, the detection rating measures the ability of the system, using current design controls, to (a) detect failure before it occurs, or (b) control the effects of failure. Our detection rating, listed in Table IV, is different than a typical detection rating because it is difficult to detect failure in the communication system before it occurs. Consistent with our teleological approach, the ratings are determined from the perspective of the system as it performs a mission; in particular, they are determined by the timing of failure detection, since the sooner it is detected, the sooner it can be corrected. Timing is important when the goal is to achieve complete coverage of an area. The rating is lowest if the formation can detect the failure and correct it during the mission. It increases if the failure can be detected but not

corrected by the formation, although this is mitigated if the failure can be communicated to the ship. If the AUVs can detect the failure but not communicate with the ship, the failure would earn a Medium rating. The rating is High if the failure can only be detected during analysis of the data after the mission, and Very High if the failure would never be detected.

TABLE IV Detection Rating		
Score	Rating	Description
1	Very Low	Detected by AUVs but can be corrected during mission
2	Low	Detected by AUVs, cannot be corrected, but can be communicated to the ship
3	Medium	Detected by AUVs, cannot be corrected, and cannot be communicated to the ship
4	High	Not detected by AUVs, but only detected by the ship when AUVs return
5	Very High	Never detected

The occurrence, severity, and detection ratings are multiplied to determine the RPN, which is a quantitative, overall measure of potential system failures. Design responses are directed at those potential failures with the highest RPN. After the recommended actions have been taken, the RPN is recalculated to measure the effectiveness of those actions and determine if any further actions are warranted. For this DFMEA, no modifications were made to the RPN formula.

IV. DFMEA TABLE

We have developed DFMEA tables in detail and are currently using them in designing the communication system for our fleet of AUVs. In Fig. 3 we provide the DFMEA that records the intentional pass for the sender AUV in our system. The occurrence ratings for the “Doesn’t Send When Should” and “Sends When it Shouldn’t” modes are low because we currently have a strict communication protocol that determines a time when each vehicle can communicate. This strict protocol limits AUV cooperation, though, and this may force us to modify the protocol; if we do, the occurrence rating will likely increase. These modes also have a detection rating of three because the logic has not yet been developed to support formation-to-fleet communication.

The “Sends When Should but Wrong” mode has a very high RPN. In the current simulations it is easy for a vehicle’s memory to become incorrect, causing it to send an incorrect message. This can cause gaps in the search pattern, which gets the highest severity rating because it is associated with missed mines that could result in loss of equipment or personnel. In addition the vehicles currently have no method of determining what cells have been scanned, and if a vehicle is lost, the personnel on the ship will have no way of determining exactly what cells had been covered. The initial part of our solution to this problem is to have each of the AUVs in the formation keep a coverage map that tracks missed cells [14]. This reduces the RPN from 100 to 60. The next step is to have the formation redirect vehicles to cover the missed areas.

V. CONCLUSION

If MCM involving AUVs is to be successful, AUVs must be designed to detect and repair error during a mission. In particular, given the vehicular and contextual constraints on communication, mission success will require detection and repair of communication error. We recommend adopting a teleological design approach that addresses error systematically under the aspect of the ultimate design goal, viz., the goal of producing autonomous agents capable of interactive communication. We structure our engagement with error by modeling it as *failure* with the help of a modified Design Failure Mode and Effects Analysis. This framework enables us

to identify, rate, and respond to error in a way that is conducive to overall design success.

ACKNOWLEDGMENTS

The authors acknowledge the Office of Naval Research, which has supported the work under the award “Developing Fleets of Autonomous Underwater Vehicles” No. N00014-05-1-0674. This publication was also made possible by NIH Grant No. P20 RR016454 from INBRE Program of the National Center for Research Resources.

[illegible]

Fig. 3. Example DFMEA—the intentional pass for the sender AUV in the AUV communication system.

REFERENCES

- [1] J. S. McCue, R. D. Blidberg, and M. J. Carter, "An investigation into protocols for underwater communication," in *Proceedings of the 7th International Symposium on Unmanned Untethered Submersible Technology* (AUV 91), Lee, N.H.: Autonomous Undersea Systems, 1991.
- [2] A. G. Rajala and D. B. Edwards, "Vehicle replacement in AUV formation," in *Proceedings of the 14th International Symposium of Unmanned Untethered*
- [3] A. G. Rajala, D. B. Edwards, and M. O'Rourke, "Collaborative behavior for vehicle replacement in AUV formations," to appear in the Proceedings of the 2005 ASME International Mechanical Engineering Congress and Exposition, Orlando, Florida, November 2005.
- [4] D. Welling and D. B. Edwards, "Multiple autonomous underwater crawler control for mine reacquisition," to appear in the Proceedings of the 2005 ASME International Mechanical Engineering Congress and Exposition, Orlando, Florida, November 2005.

- [5] M. Greaves, H. Holmback, and J. Bradshaw, "What is a conversation policy?" in *Issues in Agent Communication*, F. Dignum and M. Greaves, eds., Heidelberg: Springer-Verlag, 2000, pp. 118-131.
- [6] E. H. Turner, S. G. Chappell, and S. A. Valcourt, "COLA: A language to support communication between multiple cooperating vehicles," in *Proceedings of the 1994 Symposium on Autonomous Underwater Vehicle Technology*, pp. 309-316, 1994.
- [7] *DFMEA Guidelines*. UI Mechanical Engineering Senior Capstone Design. 14 March 2006. <http://seniordesign.engr.uidaho.edu/Course_Docs/Phase4/DFMEA.pdf>.
- [8] *TE DFMEA Presentation*. Transparent Engineering. 14 March 2006. <<http://www.transparentengineering.com/Reliability/TE%20WEB%20DFMEA%20presentation%20-%20OPG%20050%20005%20-%20rev%201.2.pdf>>.
- [9] E. H. Turner, "A hierarchy of AUV communication," in *Proceedings of the 9th International Symposium on Unmanned Untethered Submersible Technology* (AUV 95), Lee, N.H.: Autonomous Undersea Systems, 1995.
- [10] C. Goodwin and J. Heritage, "Conversation analysis," *Annual Review of Anthropology*, vol. 19, pp. 283-307, 1990.
- [11] C. Shannon and W. Weaver. *A Mathematical Model of Communication*. Urbana, IL: University of Illinois Press, 1949.
- [12] F. Dretske. *Knowledge and the Flow of Information*. Cambridge, Mass.: The MIT Press,
- [13] D. Dennett, "Intentional systems," in *Brainstorms*, Cambridge, Mass.: The MIT Press, 1978.
- [14] A. Rajala and D. Edwards, "Allocating AUVs for Mine Map Development in MCM," To Appear in *Proceedings of Oceans'06 Asia Pacific IEEE*, Singapore, May 16-19, 2006