Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

# Enabling Intra-Robotic Capabilities Adaptation Using An Organization-Based Multiagent System

Eric Matson Scott DeLoach
Multiagent and Cooperative Robotics Laboratory
Department of Computing and Information Sciences, Kansas State University
234 Nichols Hall, Manhattan, Kansas 66506, USA
{matson, sdeloach}@cis.ksu.edu

*Abstract: In harsh or dangerous environments, robots can lose function in multiple sensors and effectors over the mission, thus reducing their overall capability. If the capabilities provided by these sensors/effectors are necessary for mission completion, having an adaptive system that can overcome these losses is critical to mission accomplishment. In this paper, we propose a solution using multiple agents, organized as a team, to give robots the ability to adapt and overcome sensor/effector loss. When a sensor/effector is lost, the team can reorganize to provide the robot highest operational utility, given its current capabilities. The robot can adapt to such losses by substituting other sets of sensors/effectors to provide the best overall capability. While the robot may operate at a lower level of effectiveness, it will be able continue its mission, if possible.*

*Keywords: Organization, Capability, Multiagent Systems*

## I. INTRODUCTION

Humans possess five senses in which to interpret, communicate with and reason about their physical environment. If a human loses sensor function, such as the loss of sight, the other sensors in their body adapt by becoming more attuned to other environmental stimuli, allowing that person to, at least partially, compensate for the loss of function. This sensory adaptation is found not only in humans but also with other animal species. Other species also have the ability to compensate for lost capability, through tradeoff to other sensory types or by capability adaptation. In this paper, we introduce an organization-based multiagent system model to capture the adaptive abilities found in animal species and extend it to robots.

A common use for robots is conducting work in environments deemed too dangerous for humans. Examples include military operations such as reconnaissance over enemy held territory, clearance of land mines, and space travel/exploration [1]. In such a dangerous environment, it is probable that physical robotic capabilities, such as sensors, will suffer damage, either limiting their functionality or making them completely inoperable. Using the model of human and animal self-organization [2], we examine sensor capability tradeoff and sensor compensation using organization-based algorithms integrated into a multiagent system (MAS) architecture.

In our research, we have developed an organization-based MAS (OMAS) model that can be applied to the problem of sensor capability loss in robots [11]. As one sensor becomes unusable or loses function, another sensor is allocated to the organization and the entire set of sensor agents potentially reassigned to new tasks for to insure a fault tolerant system capable of continuing to accomplish its goals.

The goal of this research is to show the viability of applying organizational models and MAS for use in robotics. Specifically, we want to show that the resulting Organization-based multiagent systems are a highly useful and functional alternative to traditional teamwork schemes and formalisms [3, 4]. Complimentary work in this area has proposed the use of networked robotics without a self-reorganizing multiagent concept [5]. Our research takes into consideration fault tolerant systems and architectures that deal with detecting and handling sensor failure and faults [6], and calibration of sensors to adapt to unknown environmental conditions [7]. Our model tolerates faults by managing the available hardware sensors as a group, focusing on managing their entire set of capabilities instead of simple "brute force" approach to sensor switching in cases of failure.

This paper defines our organization model in Section II. In Section III, a simple multiagent approach is defined and then extended by integrating our organization model. Section IV describes the application of the OMAS model to a single robot to implement capability adaptation. Section V describes the results from our implementation evaluation while Section VI concludes by describing further OMAS research.

## II. ORGANIZATION MODEL

To implement teams of autonomous, heterogeneous agents, we created an *organizational model*, which defines and constrains the required elements of a stable, adaptable and versatile team. While most people have an intuitive idea of what an organization is, there are no standard definitions. However, in most organizational research, organizations have typically been understood as including *agents* playing *roles* within a *structure* in order to satisfy a given set of *goals*. Our proposed *organizational model* (O) is contains a structural model, a state model and a transition function.

$$O = \langle O_{structure}, O_{state}, O_{trans} \rangle$$

Fig. 1 shows the combined structural and state models using standard UML notation. The *structural model* includes a set of goals (*G*) that the team is attempting to achieve, a set of roles (*R*) that must be played to attain those goals, a set of capabilities (*C*) required to play those roles, and a set of rules or laws (*L*) that constrain the organization. The model also contains static relations between roles and goals (*achieves*), roles and capabilities (*requires*), and individual roles (*related*).

This work is sponsored by the Air Force Office of Scientific Research (AFOSR) under grant number F49620-02-1-0427.

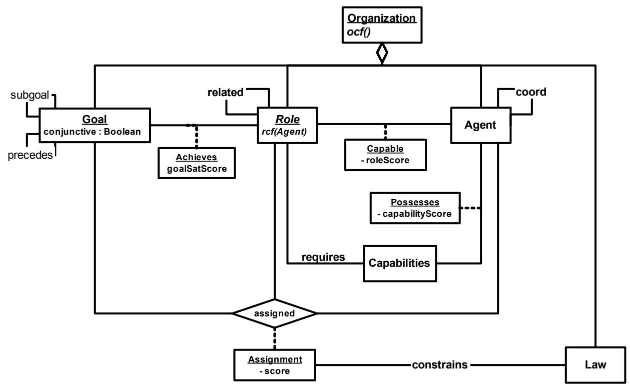


Fig. 1.Organizational Model

Formally, we model the organization structure as a tuple.

$$O_{structure} = \langle G, R, L, C, achieves, related, requires\rangle$$

where

achieves: R, G $\rightarrow$ [0 .. 1]
related: R, R$\rightarrow$ Boolean
requires: R, C $\rightarrow$ Boolean

The team *goals* include the goal definitions, goal-subgoal decomposition, and the relationship between the goals and their subgoals, which are either conjunctive or disjunctive. *Roles* define parts or positions that an agent may play in the team organization. In general, roles may be played by zero, one, or many agents simultaneously while agents may also play many roles at the same time. Each role requires a set of *capabilities*, which are inherent to particular agents and may include sensor capabilities (sonar, laser, or video, etc.), actuator capabilities (movement type, grippers, etc.), or computational capabilities (processing power, algorithms, communications, etc.). Robots are unique in the area of capabilities versus software agents; robot's physical capabilities may improve or degrade over time, which can often cause the team to reorganize. *Organizational rules* are used to constrain the assignment of agents to roles and goals within the organization. Generic rules such as "a agent may only play one role at a time" or "agents may only work on a single goal at a time" are common. However, rules are often application specific, such as requiring particular agents to play specific roles. The structural model relations define mappings between the structural model components described above. A role that can be used to satisfy a particular goal is said to *achieve* that goal, while a role *requires* specific capabilities and may work directly with other roles, thus being *related* to those roles. *Achieves* is modeled as a function to capture the relative ability of a particular role to satisfy a given goal.

The *organizational state model* defines an instance of a team's organization and includes a set of agents (*A*) and the actual relationships between the agents and the various structural model components.

$$O_{state}= \langle A, possesses, capable, assigned, coord\rangle$$

where

possesses: A, C $\rightarrow$ [0 .. 1]
capable: A, R $\rightarrow$ [0 .. 1]
assigned: A, R, G $\rightarrow$ [0 .. 1]
coord: A, A $\rightarrow$ Boolean

An agent that *possesses* the required capabilities for a particular role is said to be *capable* of playing that role. Since not all agents are created equally, *possesses* is modeled as a real valued function, where 0 would represent absolutely no capability to play a role while a 1 indicates an excellent capability. In addition, since agent capabilities may degrade over time, this value may actually change during team operation. The *capable* function defines the ability of an agent to play a particular role and is computed based on the capabilities required to play that role (see Section III). During the organization process, a specific agent is selected to play a particular role in order to satisfy a specific goal. This relationship is captured by the *assigned* function, which includes a real valued score that captures how well an agent, playing a specific role, can satisfy a given goal. When an agent is actually working directly with another agent, it is coordinating (*coord*) with that agent. Thus, the state model defines the current state of the team organization within the structure provided by the structural model.

The *organization transition function* defines how the organization may transition from one organizational state to another over the lifetime of the organization, $O_{state(n)} \rightarrow O_{state(n+1)}$. Since the team members (agents) as well as their individual capabilities may change over time, this function cannot be predefined, but must be computed based on the current state, the goals that are still being pursued, and the organizational rules. In our present research with purely autonomous teams, we have only considered reorganization that involves the *state* of the organization. However, we have defined two distinct types or reorganization: *state reorganization*, which only allows the modification of the organization state, and *structure reorganization*, which allows modification of the organization structure (and may require state reorganization to keep the organization consistent). To define state reorganization, we simply need to impose the restriction that

$$O_{trans}(O).O_{structure} = O.O_{structure} \tag{1}$$

Technically, this restriction only allows changes to the set of agents, *A*, the *coord* relation, and the *possesses*, *capable*, and *assigned* functions. However, not all these components are actually under the control of the organization. For our purposes, we assume that agents may enter or leave organizations or relationships, but that these actions are triggers that cause reorganizations and are not the result of reorganizations. Likewise, *possesses* (and thus *capable* as well) is an automatic calculation on the part of an agent that determines the roles that it can play in the organization. This calculation is totally under control of the agent (i.e. the agent may lie) and the organization can only use this information in deciding its organizational structure. Changes in an agent's capabilities may also trigger reorganization. That leaves the two elements that can be modified via state reorganization: *assigned* and *coord*. Thus, we define state reorganization as:

$$O_{trans(state)} : O \rightarrow O \tag{2}$$

where

$$
\begin{aligned}
&O_{trans(state)}(O).\ O_{struct} = O.O_{struct} \\
&\quad \wedge O_{trans(state)}(O).\ O_{state}.A = O_{state}.A \\
&\quad \wedge O_{trans(state)}(O).O_{state}.possesses = O_{state}.possesses \\
&\quad \wedge O_{trans(state)}(O).O_{state}.capable = O_{state}.capable
\end{aligned} \tag{3}
$$

## III. Organization Based Sensor Controller MAS

Our proposed solution is based on the concept of a cooperative multiagent system, or a multiagent team. Generically, the team consists of Sensor Agents, Effector Agents and Fusion Agents. *Sensor Agents* physically monitor and communicate with the hardware sensors and serve as the sensor's software interface. *Fusion Agents* understand how to fuse data captured by the Sensor Agents into information streams. *Effector Agents* use the fused information to act upon the environment. The combination of the Fusion, Effector and Sensor Agents comprises the set of all agents required to monitor, interpret and react to the environment. The simple sensor controller MAS shown in Fig. 2 is static and therefore lacks the ability to adapt to sensor loss or attrition of Sensor, Effector or Fusion Agents. If a sensor fails, the team cannot fully accomplish its mission.

The organization-based version of the system integrates our organizational model. The result is a system with the ability to alter its organization in the case of a team member loss or sub-optimal performance. An example structure of an OMAS is shown in Fig. 3 where the OMAS contains $\text{Agent}_0$ thru $\text{Agent}_n$ connected to Sensor Agent $A_0$ through Sensor Agent $A_n$. The Sensor B agents are not part of the OMAS in this description as there is no sensor failure. To fully understand OMAS, we define the foundational principles of *capability adaptation* and *capability maximization* through the formalization of basic capability concepts.

### A. Capability Formalization

So far, we have used the term capability generically. However, we must define it more precisely before moving on. A capability's existence is based on the collective sense in which it is viewed. To specify this we further define capabilities in relation to agent and roles that exist within a self-reorganizing multiagent team. As described above, an agent *possesses* specific capabilities while roles *require* particular capabilities, each with specific scores.

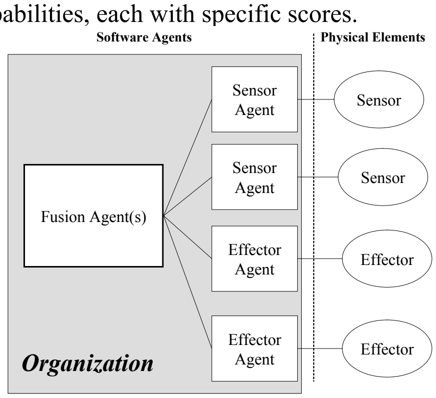


Fig. 2.Multiagent System (MAS)

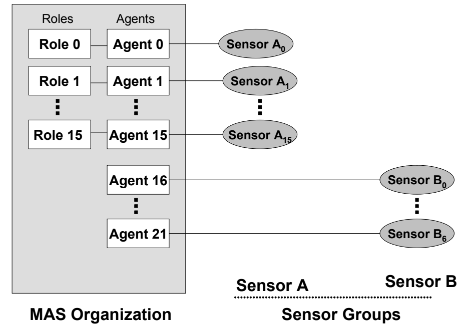


Fig. 3. Organization-based MAS

The *capability set of an agent*, $C_a$, varies from the empty set, if the agent possesses no capability, to a complete set of the capabilities that the agent intrinsically possesses. Normally even a simple agent has multiple capabilities.

$$C_a(a) = \{c \mid possesses(a,c) > 0\} \tag{4}$$

Likewise, the *capability set of a role*, $C_r$, is the set of capabilities required to play that specific role. All non-trivial roles must have at least one capability in order to accomplish some task or goal.

$$C_r(r) = \{c \mid requires(r,c)\} \tag{5}$$

The capability of an agent, *a*, to play a specific role, *r*, is computed by the role capability function *rcf*, which is part of the role definition. If an agent does not possess a required capability, then the agent has no capacity to play that role (*r.rcf(a) =0)*. Thus, the *capability score* of an agent playing a particular role is defined as

$$capable(a,r) = r.rcf(a) \tag{6}$$

The agents that form a team have a collective capability. Similarly, the set of roles required to achieve the overall organizational goal also have a set of required capabilities. We define these as *team capabilities*, $C_A$, and *required capabilities*, $C_R$.

$$C_A(O) = \{c \mid \exists a : A \bullet c \in Ca(a)\} \tag{7}$$

$$C_R(O) = \{c \mid \exists r : R \bullet c \in Cr(r)\} \tag{8}$$

To form a viable organization, these sets must be minimally overlapping such that the capabilities required are contained in the capabilities available from the agents such that $C_R(O) \subseteq C_A(O)$.

### A. Capability Adaptation

*Capability adaptation* occurs when one or more sensors are substituted for another sensor during reorganization. Adaptation is the transition the organization must realize to include the new sensors and use them to carry out the organization's mission. The organization allows an adaptation to a new sensor (or sensors) that can substitute at least some percentage of its predecessor's capability. The *possesses* value for each sensor is context dependent. For example, consider three types of sensors: sonar, tactile bump and infrared. At an abstract level, each sensor type can sense objects in the robot's task environment, but at differing levels of capability. In one situation, the bump sensor may provide higher capability, whereas in another situation the sonar or infrared sensor may provide a higher capability.

### B. Capability Maximization

Our research assumes the organization strives to operate at all times using the optimal configuration. To achieve the optimal organization, the *assignment* of agents to roles and goals, must be maximized. If the organization has a choice in which agents play which roles, it should generally choose the more capable. In terms of robots, the organization will opt to employ the most capable sensors given the current situation. Ideally, an organization will select the *best* set of assignments to maximize its ability to achieve its goals, which requires maximizing its *organizational capability score*, Os, given by

$$Os = \sum_{\forall a,r,g} assigned\ (a,r,g) \tag{9}$$

where *assigned(a,r,g) = 0* if that agent is not assigned to play a specific role to satisfy a goal.

### C. Example Scenario

To demonstrate capability adaptation we present a simple multiagent team scenario. The team organization has one overall goal, $sense_{0\text{-}270}$, which is decomposed into three subgoals – $sense_{0\text{-}90}$, $sense_{90\text{-}180}$, and $sense_{180\text{-}270}$. There are four agents (*sonar1*, *sonar2*, *sonar3*, *bump1*), three roles (*sensorA*, *sensorB*, *sensorC*), and four capabilities ($detect_{0\text{-}90}$, $detect_{90\text{-}180}$, $detect_{180\text{-}270}$, $detect_{270\text{ - }360}$). Thus, the organization is defined using the following sets.

$$\begin{aligned}
&G = \{sense_{0-90^\circ}, sense_{90-180^\circ}, sense_{180-270^\circ}\}\\
&A = \{sonar1, sonar2, sonar3, bump1\}\\
&R = \{sensorA, sensorB, sensorC)\\
&C = \{detect_{0-90^\circ}, detect_{90-180^\circ}, detect_{180-270^\circ}, detect_{270-360^\circ}\}
\end{aligned}$$

The capabilities required for each role are as follows.

$$\begin{aligned}
&C_r(sensorA) = \{detect_{0-90^\circ}\}\\
&C_r(sensorB) = \{detect_{90-180^\circ}\}\\
&C_r(sensorC) = \{detect_{180-270^\circ}\}
\end{aligned}$$

Initially, the capabilities of the four agents are:

$$\begin{aligned}
&C_a(sonar1) = \{detect_{0-90^\circ}\}\\
&C_a(sonar2) = \{detect_{90-180^\circ}\}\\
&C_a(sonar3) = \{detect_{180-270^\circ}\}\\
&C_a(bump) = \{detect_{90-180^\circ}, detect_{180-270^\circ}\}
\end{aligned}$$

In this case, more than one possible organization state satisfies the overall system goals. Assuming sonars have a higher capability score than bump sensors for obstacle detection, the organization chooses sonars over bump sensors and the initial assignment set is as follows.

$$\begin{aligned}
&(sense_{0-90^\circ}, sensorA, sonar1)\\
&(sense_{90-180^\circ}, sensorB, sonar2)\\
&(sense_{180-270^\circ}, sensorC, sonar3)
\end{aligned}$$

Fig. 4. Nomad Scout Robot

In this case, sonar1 is assigned to play the role of sensorA, sonar2 is assigned the role of sensorB, and sonar3 is assigned the role of sensorC. If either sonar2 or sonar3 fail, the team could reorganize and replace either with bump1. If sonar3 failed, the new organization assignments would be as follows.

$$\begin{aligned}
&(sense_{0-90^\circ}, sensorA, sonar1)\\
&(sense_{90-180^\circ}, sensorB, sonar2)\\
&(sense_{180-270^\circ}, sensorC, bump1)
\end{aligned}$$

## IV. OMAS Implementation

To evaluate the OMAS, we simulated specific scenarios using a robot based on the Nomad Scout robot, as shown in Fig. 4. The Scout is a simple robot, but it is sufficient for this research.

An important measure of a robot is its physical abilities, each with a specific set of capabilities to play a role within an organization. Whereas a robot is defined by its computational and physical characteristics and capabilities, we used the common sonar and bump sensors to evaluate adaptation and capability tradeoff.

The Nomad Scout robot has sonar and tactile bump sensors. The sonar ring is a Sensus 200 consisting of 16 Polaroid 6500 sonar ranging modules fixed in 22.5º increments in a full 360º configuration. The Polaroid 6500 module can accurately measure distances from 6 inches to 35 feet, ± 1%. There are six bump sensors configured on the front and rear arcs of the robot. The bump sensors are tactile, so to physical contact must be made with a physical object to trigger a response. The bump sensors, unlike the sonar, do not provide a 360º range of detection [12]. A graphical comparison of the sonar and bump sensor configurations is shown in Fig. 5, which shows that sonar 3-5 and 11-13 cannot adapt to bump sensors because there is a "dead area" with the bump sensor configuration. This indicates that if any sonar, in this range go out, there cannot be an adaptation, which negates the possibility of successful reorganization.

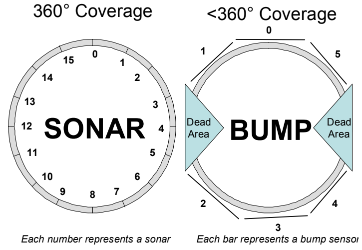


Fig. 5. Comparison of Sensor Coverage

Table 1. Capability Scores

| Agents\Roles | A | B | C | D | E | F | G | H | I | J | K | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sonar 0 | 1.0 | | | | | | | | | | | | | | | |
| Sonar 1 | | 1.0 | | | | | | | | | | | | | | |
| Sonar 2 | | | 1.0 | | | | | | | | | | | | | |
| Sonar 3 | | | | 1.0 | | | | | | | | | | | | |
| Sonar 4 | | | | | 1.0 | | | | | | | | | | | |
| Sonar 5 | | | | | | 1.0 | | | | | | | | | | |
| Sonar 6 | | | | | | | 1.0 | | | | | | | | | |
| Sonar 7 | | | | | | | | 1.0 | | | | | | | | |
| Sonar 8 | | | | | | | | | 1.0 | | | | | | | |
| Sonar 9 | | | | | | | | | | 1.0 | | | | | | |
| Sonar 10 | | | | | | | | | | | 1.0 | | | | | |
| Sonar 11 | | | | | | | | | | | | 1.0 | | | | |
| Sonar 12 | | | | | | | | | | | | | 1.0 | | | |
| Sonar 13 | | | | | | | | | | | | | | 1.0 | | |
| Sonar 14 | | | | | | | | | | | | | | | 1.0 | |
| Sonar 15 | | | | | | | | | | | | | | | | 1.0 |
| Bump 0 | 0.2 | | | | | | | | | | | | | | | |
| Bump 1 | | | | | | | | | | | | | | | 0.2 | 0.2 |
| Bump 2 | | | | | | | | | | 0.2 | 0.2 | | | | | |
| Bump 3 | | | | | | | | | 0.2 | | | | | | | |
| Bump 4 | | | | | | | 0.2 | 0.2 | | | | | | | | |
| Bump 5 | | 0.2 | 0.2 | | | | | | | | | | | | | |

To enable reorganization, we defined the initial capability scores (*capable*) for each agent in the system (each one corresponding to a specific sensor) on the Nomad. The initial capability scores for each agent and each role (A-Q) are shown in Table 1. Empty table entries correspond to a 0 score.

In designing a simulation to test the reorganization of the robot's sensor capabilities, several cases were considered. The first case is for the robot to find impediments in a room. The second case is obstacle avoidance. Case 3 is avoiding obstacles where there are other robots in the room using the same sonar module. We used the first case to consider organizational adaptation of sonar to bump sensors. As previously stated, the adaptation of these two sensor types is situation dependent. This case involves the robot searching a large area for impediments. Because of the size of the area, the sonar agents have a higher *rcf* score (1.0) due to their ability to sense up to 35' in all directions; bump sensors must come into direct contact thus lowering their *rcf* score (0.2). Because the sonar had a higher *rcf* score and could play all 16 roles, the initial assignment of roles to agents consisted solely of sonar agents as shown in Fig. 6. To limit the number of active agents, the organization allows only one agent at a time to play each of the 16 roles.

When a sonar sensor fails, its agent can no longer play its assigned role and the MAS must reorganize. If another (bump) agent has the required capability, it will be selected to play the appropriate role. Even if successful, the reorganization will reduce the team's overall capability score, as defined in Equation 9, due to the lower capability scores of the replacement bump sensor agents. Another constraint, shown in Fig. 5, are the dead areas of the bump sensor configuration, which indicates the robot has blind spots on each side due to the lack of bump sensors. If any of the sonars 3 - 5 or 11 - 13 fail, a new organization cannot be found and reorganization fails. An example of a valid reorganization is shown in Fig. 6 and Fig. 7, where sonar 15 fails, thus causing the Sonar15 agent playing role *Q* to fail. In this case, the MAS will reorganize and select sensor *Bump1* to play role *Q* as shown in Fig. 7.

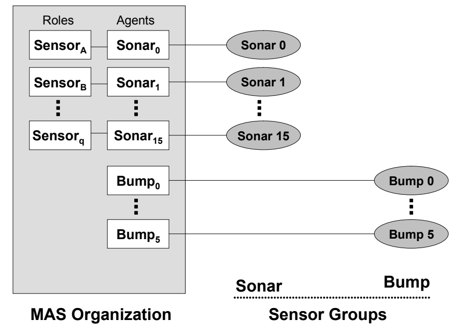


Fig. 6. Initial MAS Organization

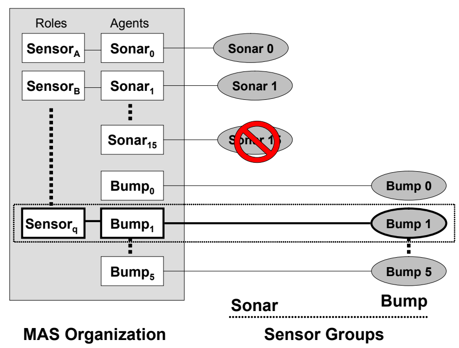


Fig. 7. Re-organized MAS

To evaluate the OMAS, we developed a Java simulator to provide large sample statistics and results over thousands of executions. We measure the effects of the executions with differing predicted sensor failure rates also taking into consideration that some sonar failures do not allow for successful reorganization. The *success* of the experiments is measured by whether the OMAS successfully reorganizes when failures occur. The baseline scenario was used to show that the OMAS could initially organize without issue with 0% sonar failure rate. Then, the failure rates where then increased in 1% and 5% increments, which were then compared to evaluate the validity of the OMAS organization. In each percentage increment, 1000 to 10,000 executions were run, incrementing by 1000, to create a large sample size and therefore add validity to the results.

## V. Results

In this section, we show how an organization-based MAS is capable of overcoming sensor incapacitation or loss within a dangerous or hazardous environment. In the base case, a 0% failure rate, the success of initial organization was equal to the number of executions. Simply stated, the OMAS correctly organized under conditions where no sonar failed.

Our initial expectation was that the results would linearly approximate the number of available sensor adaptation possibilities. Because 6 of the 16 sonars cannot adapt, intuition indicates that model can successfully reorganize 62.5% of the time and fail 37.5%. Fig. 8 illustrates the successes to executions ration where the sonar failure rate is stratified by 1% increments. It shows a nearly linear relationship. Fig. 9 illustrates the successes to executions ration with a 5% failure increment and still maintains a nearly linear trend.

Our research results match our expectation that the OMAS would successfully reorganize when a valid adaptation was possible. Thus, use of our organization model produced an adaptive system in which sensor adaptation was used to overcome loss of capability.

The single robot used in this paper is a simplified application of our organization-based model and does not demonstrate the full power of the model. Examples of more complex implementations will include *cascaded* capability adaptations where several types of adaptations may exist. For instance, a sonar failure could be replaced by a laser, a tactile sensor, or a combination of both. The capability-based nature of the model allows the automatic computation of many adaptations beyond simple replacements. In addition, different roles designed to satisfy the same goal may be used to compensate for sensor/effector failures by using existing capabilities in new ways, e.g., rotating the robot to cover areas usually covered by non-working sonar.

Although this paper deals with intra-robotic capability, our organization model can also be applied to multiple robots working as a team where the sensor/effector capabilities of one robot can fail over to another robot in a complete or partial manner. For example, a robot responsible for lifting and carrying objects can compensate for another robot.

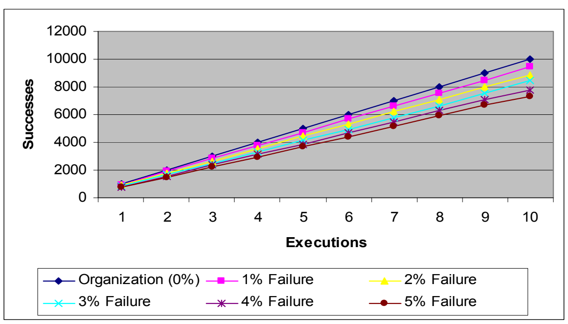


Fig. 8. 1% Increment Failure Rates

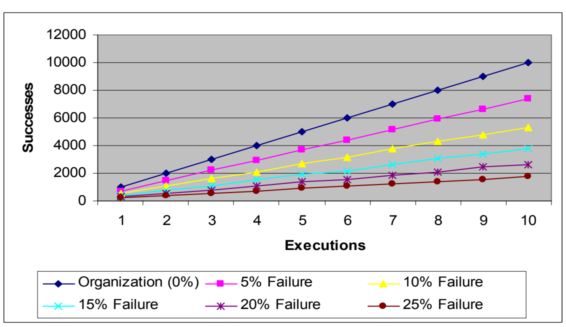


Fig. 9. 5% Increment Failure Rates

## VI. FUTURE WORK

This work is part of a larger effort to more fully define the usefulness of an organizational approach to constructing multiagent and cooperative robotic systems. In the near future, we plan to develop systems where capabilities can be shared across cooperative robots working as part of a team. The expansion of the research to cooperative robotic teams will allow capability failures in one robot to be compensated for by another robot within the same organization. A secondary area of interest is adding additional sensor instances to incrementally increase the complexity of the agent teams and to challenge the ability of the OMAS model. The addition of new sensor types and thus more, different types of agent capabilities will allow us to more fully evaluate the scalability of the organizational model and the effectiveness of our organizational reasoning techniques.

## REFERENCES


[1] Pederson, L., Kortenkamp, D., Wettergreen, D., Nourbakhsh. A Survey of Space Robotics, Proceedings of the Robosphere 2002 Workshop, NASA Ames Research Center, Moffett Field, California, November 14-15, 2002.

[2] Shen, W., Chuong, C., Will, P. Simulating Self-Organization for Multi-Robot Systems. International Conference on Intelligent and Robotic Systems, Switzerland, 2002.

[3] Tambe, M. and Zhang, W., Towards Flexible Teamwork in Persistant Teams. Second International Conference on Multi-Agent Systems, 1996.

[4] Far, B.H., Hajji, H., Saniepour, S., Soueina, S.O., Elkhouly, M.M., Formalization of Organizational Intelligence for Multiagent System Design. IEICE Transactions on Information and Systems. Volume E83-D, No. 4, April 2000.

[5] McKee, G., Schenker, P., Baker, D. Networked Robotics Concepts for Space Robotics Systems, Proceedings of the Robosphere 2002 Workshop, NASA Ames Research Center, Moffett Field, California, November 14-15, 2002.

[6] Roumeliotis, S., Sukhatme, G., Bekey, G., Sensor Fault Detection and Identification in a Mobile Robot, Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '98), Victoria, Canada, Oct. 13-17, 1998, pp.1383-1388.

[7] Soika, M. Grid Based Fault Detection and Calibration of Sensors on Mobile Robots. Proceedings of the 1997 IEEE International Conference of Robotics and Automation (ICRA '97). Albuquerque, New Mexico. April 1997.

[8] Matson, E., DeLoach, S. Organizational Model for Cooperative and Sustaining Robotic Ecologies, Proceedings of the Robosphere 2002 Workshop, NASA Ames Research Center, Moffett Field, California, November 14-15, 2002.

[9] Cabri, G., Leonardi, L., and Zambonelli, F. Implementing Agent Auctions using MARS. Technical report MOSAICO/MO/98/001.

[10] Carley, K.M., and Gasser, L. Computational Organization Theory. In G. Weiss, ed. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge, MA, 1999.

[11] Matson, E., DeLoach, S. Capability in Organization Based Multi-agent Systems, Proceedings of the Intelligent and Computer Systems (IS '03) Conference, Information Society. Institute Jozef Stefan, Ljubljana, Slovenia, October 13-17, 2003.

[12] Nomadic Technologies, Inc. July 12, 1999. Nomad Scout User's Manual, Mountain View, CA