Robotics for Future Land Warfare: *Modular Self Reconfigurable Robots*

Craig Eldershaw, Mark Yim,
David Duff, Kimon Roufas, Ying Zhang
Palo Alto Research Center
http://www.parc.com/modrobots/

Abstract

The face of modern land warfare is changing rapidly. Defence organisations around the world must be constantly adjusting and improving just to maintain a comparative advantage over their opponents. Robotics is one particular area attracting growing interest amongst a number of countries. Most of their work is following along the conventional lines of designing specialised robots to perform specific tasks. Such robots work tend not to be multi-purpose and their performance suffers when forced to deal with different environments.

This paper proposes using a more versatile solution: modular self-reconfigurable robots. These are capable of adapting their very structure to match the task and environment at hand. Their extreme modular construction enables easy, in-the-field diagnosis and repair by untrained users. The massive redundancy inherent in such systems allows a remarkably sustained performance in the face of partial damage.

These three key benefits: flexibility, maintainability and robustness are clearly attributes which are useful to armed forces around the world. However Australia, with its unique defence requirements (due to geography, size of active forces and types of deployment), is particularly in a position to enjoy the important strategic benefits. This paper discusses some of the benefits of such modular systems and one particular experimental implementation: *PolyBot*.

1 MSRRs

A *Modular Reconfigurable Robot* (MRR) is constructed from a large number of discrete modules. Each module is capable of being mechanically (and usually electrically) connected to one or more other modules. The number of different varieties of modules in a single MRR is usually small (say less than five). While the capabilities of a single module, which may only have one active degree of freedom, are exceedingly modest, the combination can form an arbitrarily complex structure.

Not only can a robot of arbitrary complexity be achieved, but a huge number of radically different robots can be constructed with the same set of modules. The properties of the particular robot change with its form. For example one MRR might be built so as to have six appendages which serve as legs for walking on rough terrain (classic insect-style legged motion) while keeping a sensor payload stable. Another MRR—composed from the very same modules—might instead form a long thin snake, capable of crawling through cracks and up pipes for access to a denied areas. In fact, within certain reasonable constraints, a well designed set of modules can be used to construct a specialised robot for almost any purpose.

This provides the potential for cost savings at the factory level: making each of the many different robots from just one or two components, allows economies of scale to come into play. However except in as much as cost savings may be passed on, then this is not immediately useful to consumer. Where MRRs become more interesting, is when they become Modular *Self* Reconfigurable Robots (MSRRs).

In addition to the properties discussed above, the mechanical interface between any two MSRRs can be electrically actuated. I.e. the modules can connect (or *dock*) and disconnect under the robot's own control. By discon-

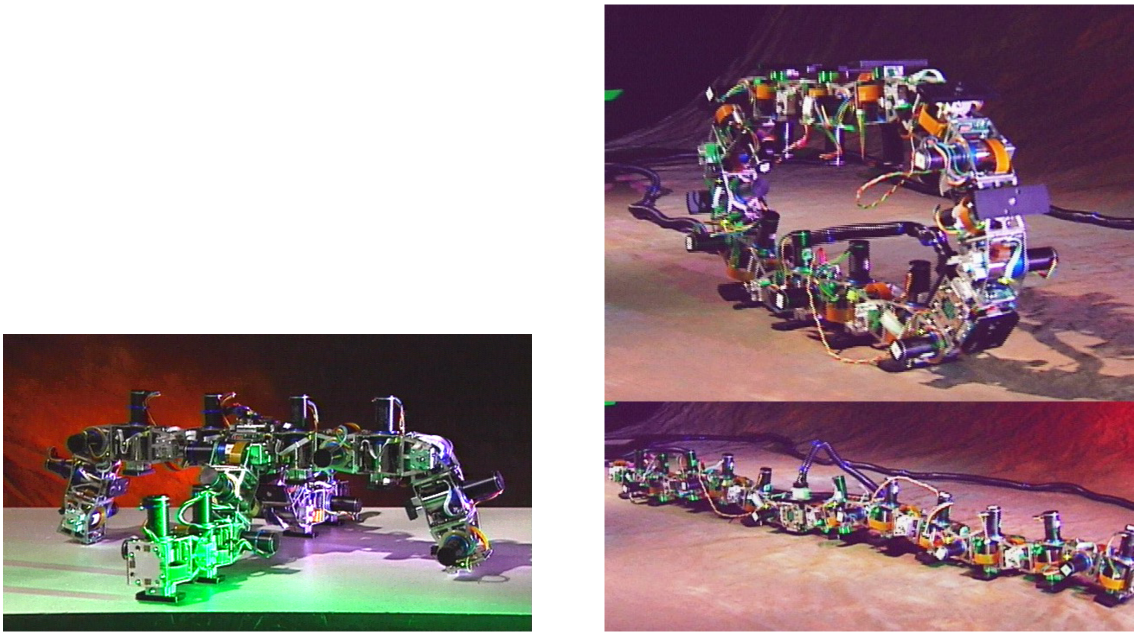

Figure 1: *A MSRR can reconfigure itself into many different forms, allowing it to chose the form most suitable for locomoting in the current environment.*

necting and re-connecting all of is modules, a MSRR is capable of completely changing its fundamental structure. It is this that allows the reconfigurable nature of the robot to really be of value.

The following section discusses the strategic advantages a MSRR can provide in land warfare—a number of which are particularly applicable for Australian forces. Section 3 describes a particular implementation of a MSRR that has been developed in conjunction with the US Defence Forces.

2 Strategic advantages

Over the last twenty years, much work has been done on developing robots and robotic technology. In this section, some of the advantages that MSRRs offer over other kinds of robots, are discussed.

2.1 Flexibility

Previous non-MSRR work (*conventional* robots), can be somewhat crudely categorised into two classes: specialised and generalised robots.

Specialised robots are by far the most common. These machines have been purpose built for performing one specific job well (for varying values of "well": fast, efficient, etc). Jobs include such things as cleaning, welding, delivering and assembling. These robots are almost invariably designed for a very specific environment. While this environment may be as broad as moving about the entire floor of a building, it is common for robots to be fixed in place in a highly structured room where all objects are positioned to fine tolerances. Such robots perform their specific jobs, in specific environments well—and in fact are optimally designed for it. However by their design, they lack any significant degree of flexibility.

While mainly only at the research stage, there do exist some more generalised conventional robots. These are specifically designed to be more flexible. E.g. helper robots for aged care which can fetch and carry and assist with walking. However even here, the repertoire of skills is restricted, and the environment is usually limited to the single floor of a relatively orderly house.

So conventional robots are usually purpose built for performing a limited number of roles in one specific environment. They will do this very well—but little or nothing else. Given its size, the Australian armed forces have found that multi-skilled personnel is not just desirable, but a necessity. Likewise, wherever possible, any piece of equipment a soldier must physically carry into the field must be multi-purpose. And so a robot, if it is to ever be worth the already burdened soldier's effort, had better be able to perform more than just one specialised role.

Soldiers rarely have the luxury of working in a single, well-defined environment. Any robot used by the Australian army, must be capable of performing its required functions in a wide variety of environments. Through rainforests or sandy deserts, maneuvering across piles of rocks or along suburban streets. The sheer variety of potential arenas that the continent of Australia provides, or that Australian troops could potentially be deployed in, is huge. These are large scale environmental variations and clearly must be planned for. However a robot small enough to be carried by a soldier faces other challenges too. Due to its size, even small objects can become obstacles, and small variations in terrain significant. Moving with speed on a flat dirt surface is quite a different skill to one that will successfully negotiate a pile of rocks or rubble [13], where each rock is larger than the robot itself. In an urban situation, crawling up a fence or drain pipe [16] is significantly different to locomoting across a lawn.

The varied (and worse: constantly changing) arena in which the modern Army finds itself, poses a particular challenge to conventional robots tuned for specific environments. An MSRR can change form, enabling it to perform multiple tasks. Such changes also allow it to adapt for locomoting through, or working in, a varied environment.

An important key to the practical use of robotics in land warfare is flexibility. While it was said in a slightly different context (discussing requirements for Australia's defence communication networks), the authors believe that the comment by Warren Harch (DSTO Land Corporate Leader) is more generally applicable: "Adaptability and reconfigurability may be more important than design for a specific purpose."[5] This is exactly the goal of *PolyBot*, the MSRR described in Section 3.

2.2 Logistics

A serious problem that faces armed forces in any extended engagement is supply logistics. In country such as Australia, with such vast areas and sparse supply bases, this is exacerbated, and that is before even considering the Australian Army's frequent overseas deployment. While it may not be a very exciting aspect of the topic, and certainly one that is almost never addressed in robotics literature, the interaction between Army logistics and a complicated new piece of equipment like a robot is an important one.

The logistics division of the Australian Defence Forces is justifiably proud of their ability to adjust rapidly to the new equipment which is constantly being rolled out. However the introduction of dozens of different specialised robots, each with a plethora of potential spare parts and complicated diagnostic procedures would be a nightmare for even the most "battle-hardened" supply officer.

The importance of rapid repair has already been noted by DSTO. The *Ultimus* project is designed for the efficient and safe clearing of land mines without the devastating corollary damage that traditional mine-sweeping methods (tanks+failing chains) can inflict. An import aspect of the project is that the sacrificial legs (hopefully the only portion of the robot to be damaged during detonation) are easily and cheaply replaced in-the-field.

By employing an MSRR, rather than the collection conventional robots, then the entire robotic parts count for the supply depots drops to just the number of different module-types in the MSRR has—in the case of *PolyBot*, this is just two. So MSRRs take the *Ultimus*'s design ideals to their logical conclusion.

In a MSRR, then the diagnostics are entirely automated, and repair trivial. If one of the modules in a MSRR fails, then this can be internally diagnosed (no single component is critical, as is discussed in section 2.3). Upon recognising this, the system will simply reconfigure while physically disconnecting the failed module. This causes the failed module to drop to the ground. Someone who has had two minutes training can then push a replacement module into any part of the robot. The MSRR will reconfigure itself again, placing the replacement wherever it belongs.

2.3 Graceful degradation

While easy support and repair of failed field equipment is clearly important, robustness against such failure in the first instance is even more critical. When in combat roles, soldiers' lives depend upon the performance of their tools. Even if issued as standard equipment, soldiers will be highly reluctant to employ any item who's reliability is in question—with good reason.

Any mechanical system, especially one subjected to the rigours of the battlefield, can potentially fatigue or break. A robot, with its many small actuators, delicate sensors and complicated electronics, is clearly a likely candidate for such failure. The authors certainly do not claim that an MSRR is any exception to this, but it does offer the unique characteristics of graceful degradation and partial self-repair.

Since a MSRR is composed of hundreds or thousands of identical modules, each with actuators and on-board processing, then massive redundancy is implicit in its design. Given the number of degrees of freedom, then the system will often be able to simply compensate for a small number of inactive modules simply at the control level. Obviously performance will be somewhat effected, but in many cases this will not be significant. In any case, continuing with reduced performance is far superior to the catastrophic failure that most conventional robots would suffer. So in this way the capabilities of the robot degrade in a "graceful" fashion.

Sometimes however, a critical module or an unfortunate combination of modules will fail. Even in this case all is not lost. Through its ability to self reconfigure, an MSRR can perform a certain amount of running repairs on itself. By reconfiguring, an MSRR can move or remove those failed modules, and replace them with other still-functioning modules from other parts of its body. To continue doing its commanded job with less modules will require some adaptation: a slight shortening of each leg perhaps (if a few modules are "borrowed" from each). However such adaptation can be made in a MSRR, something impossible with a conventional robot.

The combination of graceful degradation through adaptive control, augmented by a limited self-repair ability, means that many independent failures within the robot can be sustained without catastrophic failure of the entire system. This is a clear advantage over conventional robots.

3 PolyBot

In this section a particular implementation of a MSRR is discussed. The MSRR *PolyBot* [12], was designed and built at the Palo Alto Research Centre in California, USA. Two distinct generations and several minor variations have been created to date. A third, significantly improved generation is at this moment (May 2002) in the final stages of construction.

While several modular robotic systems have now been built at different institutions [6, 7, 8, 9, 11], *PolyBot* was one of the very first, and is still arguably the world's most sophisticated. The first generation of *PolyBot* is believed to have demonstrated a world first in being a robot which changed its own shape and locomoted in two topologically distinct ways.

While MSRRs offer many advantages, some of which were discussed in the previous section, so too do they pose many unique challenges. There exists a body of published work concerning the design, control and applications of *PolyBot*: [1, 2, 4, 10, 14, 15, 17, 18, 19]. This section gives a brief overview of this material.

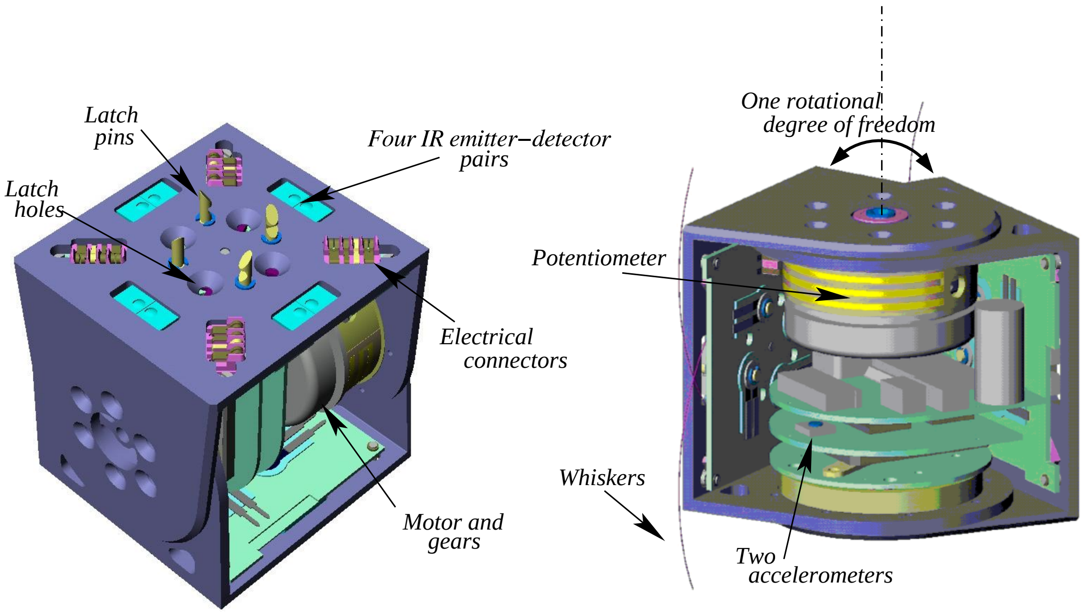

Figure 2: *A diagram showing some aspects of the third generation of* PolyBot

3.1 Hardware

The current version of *PolyBot* consists of just two module types. The *segments* are $5 \times 5 \times 5$ cm cubes with a single degree of freedom and two interconnect faces. The *nodes* are slightly larger cubes which cannot move, but have six interconnect faces.

Each interconnect face acts as a hermaphroditic connector, allowing any face to connect any other face in any of four orientations (i.e. rotations of $n.90^o$ are allowed). The mechanical linkage is accomplished through four notched pins and matching holes. A shaped memory alloy actuator controls the latch. Each face has a four times redundant set of electrical connections which carry power and communications. Force sensors at each pin allow the robot to determine what forces/torques exist between each module. [2]

The segments' single degree of freedom is rotational: the two interconnect faces can move through -90^o to $+90^o$ with respect to each other. The rotational control is provided from a brushless DC motor via a set of gears.

A directionally controllable brake (or "ratchet") exists, which allows the module to inhibit movement in a given direction. Using this brake provides a number of benefits. Chief amongst these is the drastic reduction in power consumption through being able to often turn off the motor without losing rigidity. A further benefit occurs whenever *PolyBot* is configured so as to contain one or more closed loops. By exploiting the brakes and the singularities introduced by the loop topology, near-infinite mechanical advantage may be obtained. [14]

Each interconnect face also has four infra-red emitter/detector pairs. These are used when the faces of two modules are in close proximity (e.g. when two modules are being directed to dock with each other). They allow the robot to determine the relative placement of the two modules in all six dimensions (three rotational, three translational). [10]

Except for electrical power, each module is a complete and independent system, with its own Motorolla Power PC 555 processor, 1Mb of memory and flash. There are a pair of two-axis accelerometers in each module—one in each of the two rotational halves. Limited external sensing currently exists in the form of touch sensors ("whiskers") which report contact with the ground or obstacles.

3.2 Infrastructure

The low-level communication in *PolyBot* is provided through the Control Area Network (CAN) protocol. This is fast and robust, but is not convenient for the programmer. Software architecture, both for communications and for higher level function, is an extremely important aspect of any modular robot. With its massively parallel distributed computation and large number of sensors and actuators, the programmer *must* be provided with a comprehensive set of tools if they are to work effectively. A significant amount of effort has been expended to develop such such infrastructure, leaving the developers free to deal with higher level issues. [17]

In particular, the communications layer has been augmented by the *MDCN* (Massively Distributed Communication Network) protocol. [18] This allows arbitrary length messages to be sent in a reliable fashion to any module or dynamically changing group of modules.

Another important aspect of MDCN is the routing code. CAN is a shared bus protocol which is effective for small numbers of communicating individuals. Clearly this cannot scale to the hundreds of thousands or modules that a full system would use. This problem has been resolved through partitioning the robot's communications network at each node. Special bridge code running on each node forwards packets between these sub-nets (one per face) as required. To do this MDCN first operates in a discovery mode to determine all the required local routing tables for the current configuration. After any re-configuration, these tables are automatically updated.

At a higher level, the *ASM* (Attributes and Services Model) has been developed. [19] ASM utilises MDCN and provides the developer with the means to easily control the enormously complicated system that a MSRR inevitably leads to. Access to sensors is simplified with readings (raw or processed) being made available on a push or pull basis. The actuators (motors, brakes, IR emittors) of any module can all be directed by any other. Clearly global access to sensors or actuators does not scale, but as with the partitioning of the network, this can be resolved. Individual modules can accept responsibility for controlling some logical grouping of modules (e.g. a leg), and so the overall control of the system becomes hierarchical. Each layer of the hierarchy abstracts away some of the fine detail, making scaling possible.

3.3 High-level software

Now that several generations of hardware and the supporting software infrastructure have been successfully implemented, then the higher level software development is brought into focus. Of course this is a very broad area, but the two key areas concentrated on to date have been reconfiguration planning and motion planning.

Since the fundamental attribute of a MSRR is to be able to reconfigure itself into a different shape, then software must exist to manage these changes. Determining what shape is best for a particular task and environment combination is clearly hard, but even the problem of how to move from any given topology to another is fraught with difficulties. As mentioned in the infrastructure section above, for scalability reasons it is undesirable to solve this problem as a single global reconfiguration concerning every distinct module. Rather, a hierarchy of meta-modules and other abstractions must be employed. This problem has been considered in some depth in [1, 15].

For any given configuration, a robot may have countless tasks. However one fundamental task that will always be important is locomotion. While motion planning in general has been much studied and many practical algorithms exist, the situation for a MSRR in a combat arena is very different. Firstly, the fundamental locomotive mechanism will vary from configuration to configuration (walking with legs, crawling as a snake, rolling along as a loop, climbing a fence, etc).

The other aspect, only scantily addressed the literature, is the problem of unstructured terrain. While most robots work in very precisely controlled environments with smooth floors, perpendicular walls and "neat" rectangular or circular two dimensional obstacles, this is not the world a soldier faces. It follows then, that the world faced by a robot which is deployed by a soldier is similarly "messy" and unstructured.

In unstructured environments, there is no longer a clear definition of what constitutes an obstacle. It may be possible to negotiate a route down a small cliff, but not directly up it. However by taking an indirect route it may be possible to achieve the same location at the top of the cliff that was previously unreachable. In its most extreme form, this planning requires careful direction of each foot placement (rather than the more efficient, but less robust repeated gait).

This kind of planning is computationally expensive, especially in configurations with large numbers of modules (=degrees of freedom). However as each module contains its own processor, then the total processor power available also scales with the configuration size. The difficulty then is to correctly, efficiently and robustly implement these planning algorithms on the MSRR's parallel distributed architecture. This has been investigated for *PolyBot*, and is addressed in [3, 4].

Further work currently under way concerns the planning of longer routes. Here *PolyBot* must autonomously decide where to change from configuration to configuration, so as to best negotiate each portion of terrain with its specific local characteristics.

4 Final analysis

Australia could attempt to duplicate and improve upon the conventional robotics work developed by other countries, however even a large effort will at best achieve a temporary comparative advantage. To gain a significant strategic benefit from the effort, funds and time expended, a different approach from the mainstream land warfare robotics research is called for. This paper has presented the case that modular self-reconfigurable robotics is a good way to achieve this—particularly given Australia's unique defense requirements.

MSRRs are: multi-purpose; capable of adapting to many different and changing battlefield conditions; in-field repairable; logistically convenient; and robust to partial damage. By making the strategic decision to move into MSRRs early in their development, the Australian Army stands to gain a significant foothold, giving competitive advantage that will be difficult for other countries to match. It also ensures that this technology, particularly vital for Australia's needs, is not neglected as other countries with different requirements potentially overlook it.

References

[1] A. Casal and M. Yim, *Self-Reconfiguration Planning for a Class of Modular Robots*, Proc. of SPIE, Volume 3839, 1999

[2] D. Duff, M. Yim, K. Roufas, *Evolution of PolyBot: A Modular Reconfigurable Robot*, Proc. of the Harmonic Drive Intl. Symposium, Nagano, Japan, Nov. 2001, and Proc. of COE/Super-Mechano-Systems Workshop, Tokyo, Japan, Nov. 2001

[3] C. Eldershaw, *Heuristic algorithms for motion planning*, Ph.D. Thesis, University of Oxford, 2001

[4] C. Eldershaw and M. Yim, *Motion Planning of Legged Vehicles in an Unstructured Environment*, IEEE Intl. Conf. on Robotics and Automation (ICRA), Seoul, Korea, May 2001

[5] W. Harch, *Exploiting Australia's EW Capability to Provide Information Superiority*, The Third National Convention of the Australian Chapter of the Association of Old Crows, Adelaide, 2000

[6] K. Kotay, D. Rus, M. Vona, C. McGray, "The Self-reconfiguring Robotic Molecule: Design and Control Algorithms", Algorithmic Foundations of Robotics, 1998.

[7] S. Murata, H. Kurakawa, S. Kokaji, "Self-Assembling Machine", Proc. of IEEE ICRA'94

[8] S. Murata, H. Kurakawa, E. Yoshida, K. Tomita, S. Kokaji, "A 3-D Self-Reconfigurable Structure", Proc. IEEE ICRA'98

[9] A. Pamecha, I. Ebert-Uphoff, G. S. Chirikjian, "Useful Metrics for Modular Robot Motion Planning", IEEE Transactions on Robotics and Automation, Vol.13, No.4, 1997

[10] K. Roufas, Y. Zhang, D. Duff, M. Yim, *Six Degree of Freedom Sensing for Docking Using IR RED Emitters and Receivers*, Experimental Robotics VII, Lecture Notes in Control and Information Sciences 271, D. Rus and S. Singh Eds. Springer, 2001

[11] D. Rus, M. Vona, "Self-reconfiguration Planning with Compressible Unit Modules", Proc. IEEE ICRA'99.

[12] M. Yim, D. Duff, K. Roufas, *PolyBot: a Modular Reconfigurable Robot*, IEEE Intl. Conf. on Robotics and Automation (ICRA), San Francisco, CA, April 2000

[13] M. Yim, D. Duff, K. Roufas, *Modular Reconfigurable Robots, An Approach To Urban Search and Rescue*, 1st Intl. Workshop on Human-friendly Welfare Robotics Systems, Taejon, Korea, Jan. 2000

[14] M. Yim, D. Duff, Y. Zhang, *Closed Chain Motion with Large Mechanical Advantage*, IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), Hawaii, USA, Oct. 2001

[15] M. Yim, D. Goldberg, A. Casal, *Connectivity Planning for Closed-Chain Reconfiguration*, Proc. of SPIE, Sensor Fusion and Decentralized Control in Robotic Sys. III, Vol. 4196, Nov. 2000

[16] M. Yim, S. Homans, K. Roufas, *Climbing with Snake-like Robots*, Proc. of IFAC Workshop on Mobile Robot Technology, Jejudo, Korea, May 2001

[17] Y. Zhang, K. Roufas, M. Yim, *Software Architecture for Modular Self-Reconfiguable Robots*, IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), Hawaii, USA, Oct. 2001

[18] Y. Zhang, M. Yim, K. Roufas, C. Eldershaw, *Massively Distributed Control Nets for Modular Self-Reconfigurable Robots* accepted by 2002 AAAI Spring Symposium on Intelligent Distributed and Embedded Systems

[19] Y. Zhang, M. Yim, K. Roufas, C. Eldershaw, D. Duff, *Attribute/Service Model: Design Patterns for Distributed Coordination of Sensors, Actuators and Tasks*, submitted to AAAI02 Conference