Component Oriented Development of Autonomous Mobile Robots Facilitates Interdisciplinary Design

R.D. Michelli, S.M. Scoggins, W.J. Wiseman, J.A. Janet, A.L. Walker
TMI Robotics, Inc.

Abstract

Our experience developing mobile robots with groups of undergraduates has shown that while many teams consider their design to be interdisciplinary in nature, the design is in fact fragmented across engineering disciplines. The end result is a project that aggregates various engineering disciplines instead of integrating them into a true multidisciplinary design.

We propose a component-oriented design approach, in which more project time is devoted to system functionality and less to subsystem development. A collection of mechanical, electrical, and software components can be designed or purchased ahead of time independent of a particular project's needs. These components can then be drawn upon to rapidly develop complex systems. In addition, knowledge and design decisions specific to one particular engineering discipline can be encapsulated in a modular component, allowing the entire design team to address the issues related to component integration.

We have applied this component-oriented design approach to the multidisciplinary design of autonomous mobile robots. Students applying this method have successfully developed an autonomous bipedal walking robot and a more traditional wheeled robot with ultrasonic sonar array, tactile bumper and electronic compass. These projects were designed and completed in a single semester by teams of students. Using previously developed modular actuators, sensors, amplifiers, and software agents allowed early integration of subsystems and left more time for global system design.

I. Introduction

A mobile robot is a complex system that requires multidisciplinary design. Creation of such a system requires the synthesis and cooperation of a multitude of different subsystems. For this reason, the design of a mobile robot is a topic well suited for cross-disciplinary education. It is common for separate engineering schools to collaborate on such a design project; these programs promote interaction between students from differing fields at a technical level. This interaction helps develop the ability to communicate intelligently across engineering disciplines, a skill that will be demanded of students upon entering the workplace.

We present a new design methodology for mobile robot projects at the undergraduate level: Component Oriented Design. We found that, through the use of prefabricated mechanical, electrical, and software modules, small teams of students (5-7) were able to produce some impressive mobile robots in a single semester. Of key interest to the university community is the students' focus on global system behavior rather than subsystem development, and the financial benefits of owning a set of reusable, upgradable components. In the following sections, we review problems commonly observed in undergraduate mobile robot projects, present a view of Component Oriented Design, briefly describe some of our components, and review two mobile robots designed using those components.

II. Observed Design Methodologies

Engineering programs involving mobile robot design typically fall short of their initial goal of creating a true multidisciplinary educational experience. Actual interdisciplinary work is performed at the outset of the project to create some architectural view of the final system. With such a global goal in place, the task of subsystem design is partitioned among the various engineering disciplines represented. We have observed that the subsystem design phase of a project represents the vast majority of the project timeline, as shown in Figure 1. This is caused primarily by students becoming overwhelmed by volumes of low level, detailed design work within their own area of expertise, which is

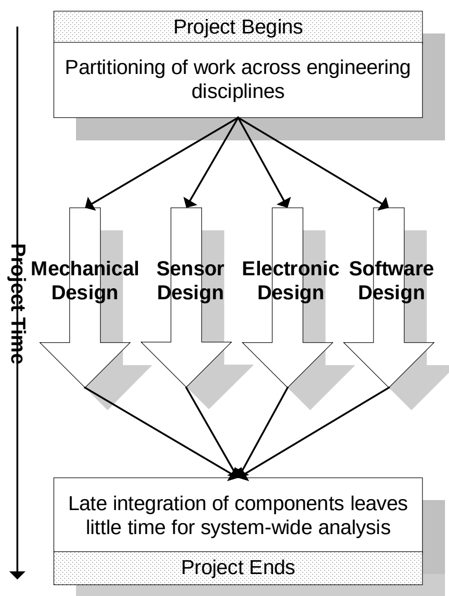

Figure 1. Observed design timeline typically found in undergraduate mobile robot projects.

sub-optimal for many reasons. The integration of subsystems is performed hastily at the end of the project, creating a loosely coupled collection of components with limited functionality. Little or no system-wide analysis is performed. More importantly, only a limited amount of interaction occurs between engineering disciplines for the majority of the project, violating the original goal of creating a multidisciplinary educational experience.

Another undesirable result is the uniqueness of each subsystem designed in the course of the project. This has negative chronological and financial implications, as future projects cannot utilize subsystems generated from prior work. For example, a student may design a drive train that, while perfectly acceptable for the current project, has no use to future projects, necessitating novel drive train designs for subsequent projects. Departments sponsoring these projects cannot achieve continuous improvements in project quality. This stems from the fact that projects must begin anew each semester. Student morale is also an issue: while they may enter into a project expecting to work on overall system behavior, they quickly become bogged down in the details of their own engineering discipline.

III. Component Oriented Design Methodology

We have applied a Component Oriented Design (COD) methodology to undergraduate mobile robot projects, which addresses the aforementioned problems. By designing and fabricating or purchasing a set of generic, modular components prior to beginning a project, we provide students with robust, easy to use, encapsulated subsystems. This directly alleviates the problem of repetitive subsystem design, allows for early subsystem integration, and, consequently, provides students with more time for system-wide design, development, and analysis.

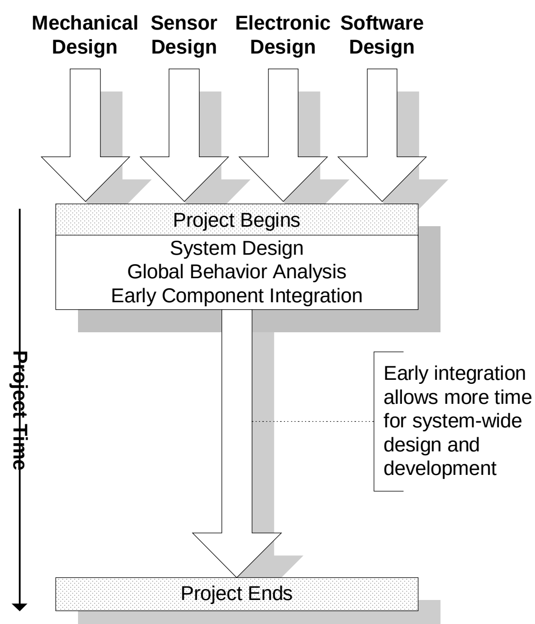

Figure 2. COD timeline as applied to mobile robot projects.

The success of a component-oriented approach to system design has been proven by the widespread adoption of the practice amongst the software development community. As internetworking technology has facilitated larger and more distributed systems, and time-to-market has become ever more critical, component-based software design has become an integral part of the software industry. Microsoft's reliance on its COM and DCOM standard for interoperable components as the foundation for all Windows application development, the widespread use of interchangeable ActiveX controls, and the ready adoption of CORBA by the enterprise computing world all attest to the power of re-usable components.

We have observed several benefits of applying COD to undergraduate mobile robot designs. Students involved in projects utilizing COD spent the majority of project time dealing with the overall system and cross-subsystem issues that required multidisciplinary interaction to solve. Their solutions can be passed on to next generation projects due to the commonality of the components. Expensive fabrication equipment or outsourcing of complex construction is eliminated, due to the preexistence of components. A department would experience significant improvement in the quality of its projects over time. The ultimate result is a multidisciplinary educational experience that not only meets the current project goals, but also provides the building blocks for future projects.

IV. Sample Components

In this section, we provide physical descriptions of a limited set of components we have utilized, along with some of our motivations for creating such components. To facilitate their use, any generic component would require the following features:

- Perform a specific function.
- Integrate all of its own subsystems (such as sensor and drive circuitry) so to be transparent, yet available if desired, to the user.
- Utilize a common interface to facilitate system integration.

Such components would render most low-level design issues solved and exist as building blocks for more complex systems. Students can become familiar with such components in a matter of days and consequently can begin system integration at the early stages of the project.

IV.1. Example Mechanical Component - Modular Rotary Actuator

The modular rotary actuator (MRA) converts motor power into high-strength rotary joint movement for applications, such as reconfigurable robotic arms and legs. The modular joint has low space and power requirements and it supports two different classes of motors to satisfy general strength-to-weight, strength-to-power and strength-to-size requirements. Strength/speed characteristics can be further refined with different combinations of interchangeable gears and motors.

The modular joint can be placed at any point as a rotary degree-of-freedom (shoulder, elbow, wrist, hip, knee, ankle, wheel, etc.) regardless of general robot configuration. Inside each gearbox is a fully adjustable cam/reference switch system to give position

Figure 3. Variants of the Modular Rotary Actuator.

references (e.g., limit and home switches). The cams can be set to activate the switches at any joint angle. In general, the modular joint represents a long-needed building block, providing new flexibility to the construction of robotic arms and legs.

IV.2. Example Electrical Component - Modular Quad-Amplifier.

The Modular Quad-Amplifier (MQA) converts 5V pulse width modulated digital signals into 12-55V driving signals. It is most typically used to drive motors from microcontrollers.

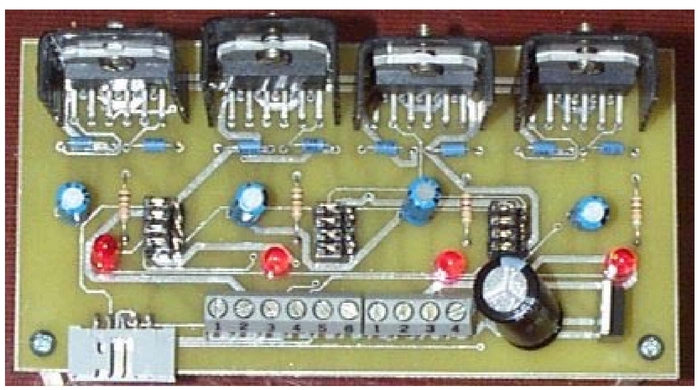

Figure 4. Modular Motor Amplifier

The MQA contains four LMD18201 amplifier circuits. They are arranged in such a way as to allow them to be used individually or to be combined in a parallel circuit containing 2, 3, or all 4 amplifiers just by moving a few jumpers. By using more than one amplifier in parallel allows for more current to be drawn by an attached device. Additionally, an on-board 5V regulator provides power for an attached microcontroller.

IV.3. Example Software Component – Ranging Subsystem Controller

The sensor subsystems on our autonomous platform, Lazarus (see Section V.2), were not only composed of modular hardware components, but also of modular software components. The sensor control modules were implemented as polymorphic types that could be readily interchanged, inserted, or removed without restarting the main control system. Through inheritance-based polymorphism and run-time binding of sensor control modules to the central controller, a wide array of sensor types could be employed without any specific knowledge of their operation in the central controller. This arrangement facilitates non-destructive upgrading of sensor capabilities while requiring no changes to be made to the central controller. Since all range sensor modules are required to implement the same interface, they can be interchanged freely – they appear identical to the central controller. Figure 5 is a UML class diagram showing the inheritance relationships amongst the various range sensors installed on Lazarus.

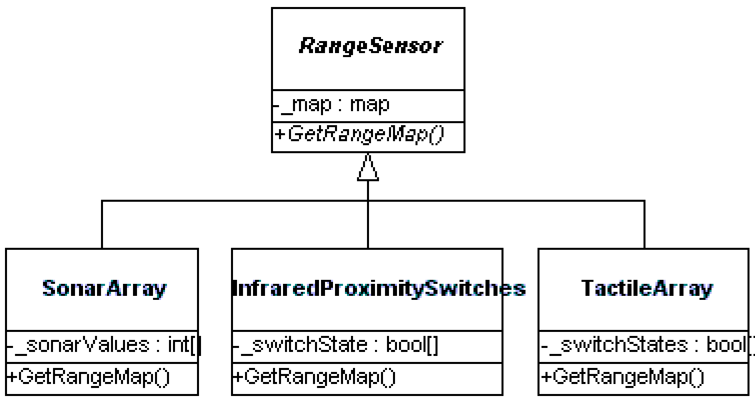

Figure 5. UML class diagram of the range sensor control classes employed on Lazarus.

V. Sample Projects

V.1. Jenner – A Statically Stable Biped Walking Robot

Jenner is a statically stable biped walking robot based almost entirely on modular components. The project was brought from conception to completion in a single semester by a team of five undergraduates and two first-year graduate students. The six-degree of freedom robot used a distributed control system based on Echelon's Neuron Chip-based microcontrollers that communicate over a twisted-pair fieldbus. Each node had a specific function, such as joint control or user interface responsibilities. The fieldbus data link allowed individual nodes to pass messages, yielding an excellent field to conduct object oriented design [1]. All six motor control nodes were identical except for a few configuration parameters, such as gear ratio and location on the robot. All six degrees of freedom were comprised of MRAs with simple modifications to suit torque and speed requirements for their particular location on the robot. This allowed the mechanical design team to concentrate on overall system design, such as joint links, walking gait, and support of the control system.

Jenner's control system also benefited from modular design. The preexistence of the MRAs gave the automatic control team access to the mechanical joints early into the project, so there was plenty of time to optimize the motion controller design to suit the performance requirements of the actuators. The control team worked closely with the mechanical team to determine which quantities should be parameterized in order to

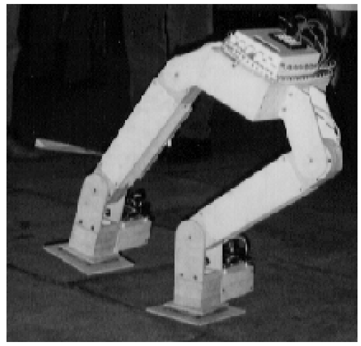

Figure 6. Jenner taking a step

create a single control algorithm that would work for all six degrees of freedom.
In fact, this project spawned a modular motion controller with integrated amplifier to complement the modular rotary actuators, which will be of great benefit to future projects. The success of the motion controller, coupled the fact that its design was completed early in the semester, allowed the teams sufficient time to concentrate on optimizing overall system performance.

The results of this project were very impressive. Jenner could repeatedly walk slowly without falling for more than two hours on a 7 Amp-hour, 12 Volt battery. The only performance failure, a fall, occurred when the battery depleted. All other times Jenner operated at the optimal performance level, which is to say the actuators acted in synch to achieve a statically stable gait. These results show what can be accomplished in a small amount of time using the COD approach. Jenner simply could not have come into existence in a single semester without the modular components. In summary, this project is a prime example of how COD can allow a design team to more effectively focus on global project goals.

V.2. Lazarus – An Autonomous Wheeled Mobile Robot

Lazarus is an autonomous mobile robot; it is equipped with the capabilities to navigate and avoid obstacles. As with Jenner, the main mechanical systems were pre-existing components and system interfacing is achieved through a network architecture based on Echelon microcontrollers. Subsystems comprising Lazarus include:

- Drive Train: TRC Labmate differential drive locomotion system
- Ranging sensors: 32-sonar ring with 1 cm precision and 10m range.
- Tactile sensor: dual-track, 16-region tactile array with 10 cm precision
- Navigation sensor: Vector 2X electronic compass
- Web server: Pentium-90 PC
- Wander agent: Software entity used to make motion decisions

Student teams have worked on Lazarus for three semesters. Each semester, more functionality is added in the form of new components. Building the robot around a fieldbus allows each new subsystem to be easily integrated into the existing structure. As a result, the components on Lazarus represent work of students from many different engineering disciplines.

Communication between components over the fieldbus creates global behavior [1]. For example, when the tactile node senses a collision, it sends a network message directly to the drive train controller, causing the robot to stop. The wander agent inspects environment maps provided by the sensor agents, picks a random direction and speed, and sends a message to the drive train to begin movement.

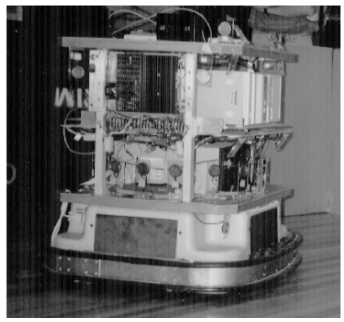

Figure 7. Lazarus operating in wander mode.

The control network allowed each unique component to be designed as a simple plug in module. This creates an open architecture for future extensions to Lazarus. As interchangeable components are added to the system, the core controller does not need to change, or even be aware of additions or removals of components. An example central controller for a mobile platform (other than the wander agent) is shown in Figure 8. An `EnvironmentMapper` module (which would replace the current wander agent) is coded to interact with `RangeSensor` (described in Section IV.3) and `SelfLocalizationMethod` objects (possibly comprised of a dead reckoner, GPS receiver, or visual cue finder). The specific ranging and localization behaviors are implemented by concrete subclasses of these types, bound at run-time to the central controller if present.

The main purpose of this project was to study the feasibility of a robot whose stand-alone subsystems could provide desired overall system performance. With the sonar, tactile, locomotion, compass and central intelligence nodes functional, the robot is capable of wandering on its own safely. Safe speeds are 0.4 m/sec and 50°/sec are limited primarily

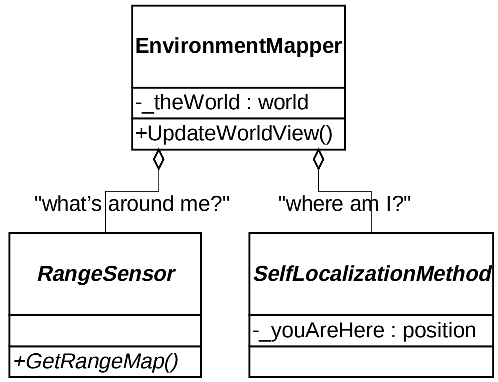

Figure 8. UML class diagram showing how a set of range sensors and a set of self localization methods can be used to generate a world view.

by the sonar subsystem. As new components are added, Lazurus' performance and functionality are expected to improve.

VI. Other Efforts in Component Oriented Mobile Robot Design

While several groups have proposed robotic systems composed of easily configurable modular components that implement the structural, sensing, or control functions of a robot, [2, 3, 4, 5], there does not exist a commercially available family of products that would allow a true component-oriented design of an autonomous robot. Although some robotics kits are available which enable construction of a complete vehicle, most of these are intended for hobbyists or as toys, (e.g. Legos Mindstorm and the Fischertechnik series). The limitations of these kits stem from their target market which does not require features such as networking of control modules, sophisticated sensor packages, or powerful processor platforms, all of which are necessary for the construction of an autonomous vehicle. Individual robot components including sensors, controllers, and mechanical structures are commercially available, but these products represent a rather ad-hoc solution to designing a modular robot that could lead to difficulties when interfacing different components. Other universities have tackled the problem of COD for their robotics courses with some success, as seen by MIT's 6.270 course which has produced several controller boards based on Motorola's eight bit microprocessor family [6]. The main problem for all of the products mentioned above is that they lack the critical features of interoperability and an open architecture. The existence of these two characteristics in a collection of components would ensure relatively simple interfacing and modularity. This would certainly help in achieving the goal of students spending most of their design time on the functionality of the system, not on the implementation of each function.

VII. Conclusions and Future Work

We have presented a design methodology for enhancing multidisciplinary interaction between engineering disciplines. Component Oriented Design techniques are used commonly by the software industry today; the methods presented are extensions of these concepts. In the future, we would like to conduct a quantitative and qualitative analysis on the benefits of COD as applied to mobile robot design. By observing two isolated teams of students attempting to complete identical projects, one team using COD and the other not, measurable differences of project quality, student satisfaction, and student benefit could be observed. By continuing this study over several semesters or years, the cost benefits, project enhancements, and other trends brought by a COD program could be empirically obtained.

References

1. Janet, J., Wiseman, W., Michelli, R., Walker, A., Wyzochanski, D., Hamlin, R. "*Applications of Control Networks in Distributed Robotic Systems*" Proc. IEEE Conference on Systems, Man, and Cybernetics, 1998. pp. 3385-3370.
2. Fernandez, J.A., Gonzalez, J. "*NEXUS: a flexible, efficient and robust framework for integrating software components of a robotic system*" Proc. IEEE Conference on Robotics and Automation , 1998. Volume: 1 , pp.: 524 -529.
3. Walter, J.A. "*SORMA: interoperating distributed robotics hardware*" Proc. IEEE Conference on Robotics and Automation , 1998. Volume: 4 , pp.: 3511 -3518 .
4. Farritor, S., Dubowsky, S., Rutman, N., Cole, J. "*A systems-level modular design approach to field robotics*" Proc. IEEE Conference on Robotics and Automation , 1998. Vol. 4 , Pp.: 2890 –2895.
5. Sanderson, A.C. "*Modular robotics: design and examples*" Proc. IEEE Conference on Emerging Technologies and Factory Automation, 1996. Volume: 2 , pp.: 460 –466.
6. Manseur, R. "*Development of an undergraduate robotics course*" Proceedings of Frontiers in Education Conference, 1997. Volume: 2 , pp.: 610-612.

RICHARD D. MICHELLI
Rich Michelli is the Director of Mechanical Engineering at TMI Robotics, Inc., a North Carolina based technology development company. He received his B.S. in Mechanical Engineering and is currently pursuing an M.S. in Mechatronics from NC State University in Raleigh, North Carolina.

SEAN M. SCOGGINS
Sean Scoggins is the Director of Software Development at TMI Robotics, Inc. He received his B.S. and M.S. in Computer Engineering from NC State University. His research interests include soft computing structures, pattern recognition, object oriented design, and distributed systems.

WILLIAM J. WISEMAN
Bill Wiseman is currently a design engineer at TMI Robotics, Inc. He received his B.S. in Weapons and Systems Engineering from the US Naval Academy in Annapolis, Maryland and his M.S. in Computer Engineering from NC State University. Before returning to graduate school, he served as a naval officer at SEAL Team FIVE for over five years.

JASON A. JANET
Jason Janet is an Adjunct Professor at Duke University. He received his B.S. in Mechanical Engineering from the University of Virginia, and his M.S. in Integrated Manufacturing Systems and Ph.D. in Electrical Engineering from NC State University. He has published over 25 research papers in the areas of robotics and soft computing. His research interests include distributed and modular robotic systems.

AARON L. WALKER
Aaron Walker is currently a design engineer at TMI Robotics, Inc. He received his B.S. in Nuclear Engineering from the Kansas State University and his M.S. in Electrical Engineering from NC State University.