A software framework for the design and support of mass market CLAWAR machines

Yiannis Gatsoulis Ioannis Chochlidakis Gurvinder S. Virk

Intelligent Systems Group, School of Mechanical Engineering, University of Leeds
Woodhouse Lane, Leeds, West Yorkshire, U.K. LS2 9JT
Email: (menig, men2ic, g.s.virk)@leeds.ac.uk

***Abstract*— A software toolset for the design of robots is described. Modular and system level design issues are discussed and criteria for the appropriate selection of software design tools are presented. Case studies of a design toolset illustrate the principles.**

I. INTRODUCTION

The research and development (R&D) of robotic systems is a complicated process and the produced systems are usually expensive. Robot systems designers prefer to use a modular approach in order to reduce development and maintenance costs as well as make their systems more flexible. However, these modules are task oriented and restrained within the scope of the particular platform. This kind of modularity is called *in–house modularity*. As a result research teams are wasting considerable amounts of time and resources in developing their own modules.

On the other hand if standard components were available in the market, developers would be able to build their prototypes faster as there will be reduced need for individual development of the components. Furthermore, the whole process is more cost effective due to the increased possibility of adopting already developed techniques, mass production and reuse of the modules and the technologies for a wide range of applications. This kind of modularity is called *open modularity* [1], [2].

Figure 1 shows an approach to open modularity R&D where the overall problem is broken down into different stages. Firstly, the application areas and the operational environments are identified. A generic list of task requirements is then formed for each. The general capabilities required by the robot are then formed, followed by a design of the specific capabilities. All components are grouped under four generic categories, namely input, processing, output and infrastructure modules. Super modules can be constructed from basic ones. All modules (basic and super) integrate with each other using the interaction space as defined by the CLAWAR community [3]; this defines six ways in which interactions can take place, namely mechanics, power, digital, analogue, databus and environment. Normally we would expect system level and module level designs to be carried out in a parallel way as shown in Figure 1. Standard protocols must also be developed for that. Assessments must be conducted at different levels of the R&D process before any real models are built and the modules are finalised for mass production and effective

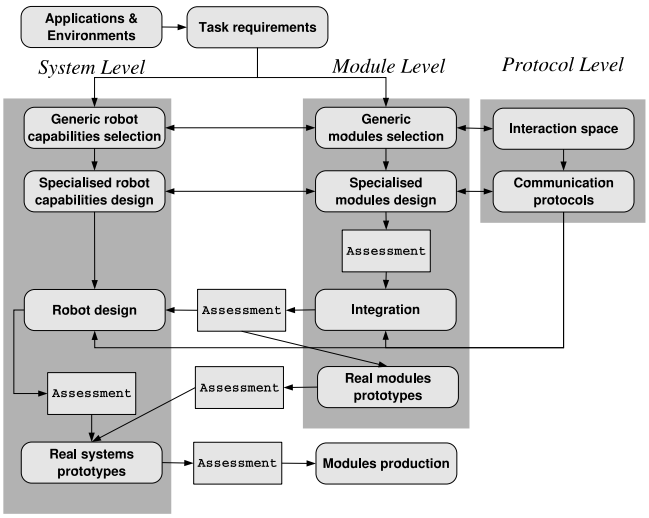

Fig. 1. Open modularity process

R&D. This is achieved by extensive testing and refining of the solutions until all aspects are achieved.

Therefore, software tools that can assist in formalising the requirements specifications in some structured manner, in analysing the problem and the systems, in designing, simulating and assessing them would be useful. However, there is a vast number of software packages with different capabilities for each and they come at different costs, from free up to thousands of pounds. As already mentioned, building a robot prototype is a complex, time consuming and expensive task and these costs are increasing. This is a major barrier to progressing the area of robotics and a step change in culture is needed in the view of the CLAWAR community. We need to start a "common thinking approach" so that we can reuse existing technology and stop wasting efforts in "re-inventing the wheel". Promoting an open modular design approach presents the most viable strategy for this to work, we need appropriate software design tools. This paper presents a software support toolset approach building on the modular work developed by CLAWAR. These tools should be selected from a menu in a fast and accurate way without any waste of time or money to realise and test concepts.

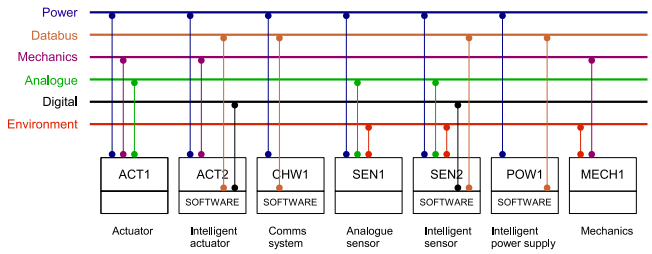

Fig. 2. Example interaction space diagram

II. Design Levels

It is shown in Figure 1 that the overall design process is separated in three distinct levels; the *system level*, the *module level* and the *protocol level*. All three levels are developed in parallel with interactions between them at different stages of their development.

The module level refers to the design and assessment issues of basic and super modules. "A module for mobile robots is described as any functionally complete device, or sub–assembly, that can be independently operated and can be readily fitted and connected to, or in combination with, additional modules to comprise a complete and functionally reliable system" [2]. Four generic categories of modules have been identified from the work and experience of the CLAWAR project [1]–[3], namely *input, output, processing and infrastructure modules*. Input modules are related with the perception and the control of the system and output modules are to allow the robot to interact with the environment and with the operators. The processing modules deal with all the computational aspects of the system, both in software as well as hardware. The infrastructure modules refer to all aspects of electromechanical support of the system, such as the mechanical materials, the locomotion system, the communications networking and the power supply and its distribution.

The protocol level deals with the design and development of communication interfaces and protocols that allow the integration of the modules themselves and in re–configurable robotic systems. This is achieved by identifying the interaction space, which describes the communications and interactions between themselves and the environment. Following the work of CLAWAR, six types of interactions have been identified; power regarding the power supply issues, mechanical regarding the physical linkages and support of the modules, analogue and digital regarding any analogue or digital flow of information respectively, databus referring to some structured and standardised representation and flow of data, and environment regarding all the interactions and influences from the operating environment. An example interaction space diagram is shown in Fig 2.

The system level concerns the issues of the complete robotic systems. These start from the high level abstract capabilities and specifications of the robots to the implementation of the complete system from the available basic and super modules. This means that the system level design has to interact with the module level design constantly. The protocol level is also assessed within this level. The complete robotic systems are assessed in simple scenarios as well as more realistic ones in their application environments. The assessments are firstly computer simulations before any real prototypes are developed, which in turn should also pass a number of assessments.

III. Software Selection Framework

It is clear that a software toolset should be adopted for the design and assessment of the issues concerned by the system, module and protocol levels. However, the vast number of available software packages makes a selection of an appropriate software framework not a trivial task. Moreover, the need for quick selection and rapid and reliable development has resulted in the proposal of several evaluation frameworks which try to provide some selection guidance to the system analysts. Most of them use some form of hierarchical organisation. One of them, defined by the ISO/IEC 9126 standard [4], has a hierarchical structure of high level characteristics that are decomposed into sub–characteristics and attributes. The high level ones are reliability, usability, efficiency, maintainability and portability. An alternative hierarchical framework considers the user, the vendor, the model and the input, the execution, the animation, the testing and the efficiency and the output of the software as the highest levels in the hierarchy [5]. Another one uses input, processing, output, support and cost as the generic categories [6]. Relative evaluation methods where the candidate software tools are compared in pairs with each other have also been used [7]. Furthermore, case studies have been applied in fields such as aerospace engineering [8], mail transfer issues [9] and structural engineering [10]. Expert systems that implement selection frameworks have been written as well [11].

In this study an evaluation framework which consists of the following criteria has been used [12].

- *Cost:* The cost of the simulator.
- *Usability:* This measures how well the design and assessment tool fulfils the simulation requirements.
- *Expandability:* This measures the how likely is the tool to be updated with new components in the near future or if it already provides this capability to the user.
- *Reusability:* This measures if a software tool can be used for the design as well as the assessment of a model; if the produced models can be used by other software tools; and if the implemented control programs can be reused with real robots.
- *Development time:* This measures how fast the new designs can be developed.
- *Efficiency:* This measures the performance and other execution facilitation of the software.
- *Visualisation:* This measures how realistic the implemented designs and environments look like.
- *Portability:* This considers if the tool can be run in the required operating systems.

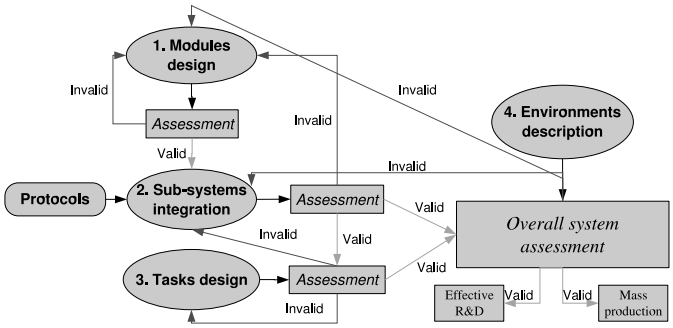

Fig. 3. Simulation cycle

- *User friendly:* This measures how easy it is for the users to learn to use the software.
- *Technical support:* This measures how likely is to get technical assistance.
- *Analysis facilitation:* This measures if the software provides any facilitation for analysing and visualising the assessment data.

IV. Case studies

In Section I the issue of the generic design of robotic systems was addressed. This requires the specification and design of the following:

- Module dynamics and statics.
- Integration of the subsystems to form complete robotic systems.
- Tasks to be performed.
- Operational environments and objects to be encountered.

A design and assessment cycle is shown in Figure 3. At the module level the basic and the super modules are normally designed and assessed both individually and together. At the system level the complete robotic systems are considered in their operational environments. A design methodology and a software toolset for the system, module and protocol levels are described and illustrated through case studies in colonoscopy and urban search and rescue applications.

A. Colonoscopy application

The case study discusses the modular design of a mobile robot for colonoscopy [13], [14]. It introduces and proposes design environments for mobile climbing robots on irregular and/or smooth terrains. There are two general categories of climbing scenarios depending on the application; the first includes all kinds of terrain vertical or horizontal, rough or smooth, while the second arises in medical applications where wet and dry, rough and smooth, and rigid and pressure–deformed surfaces need to be addressed.

The ability to handle these types of surfaces and various robot kinematic and dynamic mechanics needed to be simulated in the appropriate internal body environments. In order to visualiser the problem(s) and invent new solutions, they can be studied in the simulated set up before risking having to build anything. Applications other than medical ones can be simulated by various software packages that provide 3D design and simulation engines together with CAD capabilities for kinematic analysis. As a result of the research carried out on the software packages available, packages like Visual Nastran, Darwin2K and Yobotics were found to be suitable. Visual Nastran is felt to be the most appropriate and will be used in this case study for specific limb motion as it gives a variety of joint types and motions with various properties to match different design and simulation concepts. Player/Stage/Gazebo (PSG, system level tool) is another powerful simulation studio which includes mapping, localisations both in 2D and 3D with on board camera views, path planning and 3D world design and all these make it a useful robot design tool. Darwin2k is a combination of the PSG system level software but with module level design capabilities.

The CLAWAR design approach to start with formulating the system level requirements and breaking these into appropriate module level sub–designs is followed here. Regarding any application, the system level design should come before the modular level one. The details of how a system level approach can be realised is discussed in [12] and is used in this case study in order to continue to the modular level design. The design aspects for the modular design are focused upon here.

For the application of colonoscopy, 3D environments are vital for realising effective robot designs. All mobile robots in any environment need to sense, make some decisions and move themselves or move something. This is also the case for robot colonoscopy systems. By completing the system level design first it is possible to produce a list of the necessary modules from considering the task requirements regarding the specific application. The colonoscopy robot should be biocompatible and able to navigate within the restrictive confines of the large intestine. The main constraint apart from the biocompatibility and strength for locomotion is that it must be compact and able to stabiliser itself while navigating. In other words a mechanism must allow the robot to either grip or stick to the walls around it without rupturing them and be able to manoeuvre freely.

Having this in mind the System Level Tasks are listed below:

Working Environment: The Inner Human Body:

1) 3D, Intestine (Colon) - tight confined environment
2) Pressure–deformed surfaces
3) Danger of rupturing walls
4) Sleek/smooth walls
5) Liquid flow might be present

Tasks to be carried out:

1) Semi–autonomous in navigation
2) Allow medical examination
3) Provide visual feedback to doctors
4) Perform medical procedures (remove tumours)

Performance metrics:

1) Safe: biocompatible
2) Effective: locomotion, inspection, treatment
3) Compact

4) Reliable

Operation:

1) Remote
2) Semi-autonomous

This way of approach highlights the need for different software for the environment design and simulation. Firstly any possible CAD or robot design and simulation software could work well for applications involving rigid pipes in petrochemical plant situations [15], [16], but the introduction of soft tissue tubes in biomedical situation such as in colonoscopy exposes the need for a new type of software where pressure-deformed surfaces and tissue dynamics are included [17]. Pressure-deformed surfaces lead to the need to have specialised locomotion methods. End effectors, sensors (grippers/manipulators, pressure sensors, camera, temperature sensors, distance and collision avoidance detection sensors) and specialised mechanics are the system level design requirements. The pressure-deformed characteristic of the colonoscopy environment leads to the formulation of the following list of requirements:

- Safety: need for umbilical to retrieve device
- Soft surface dynamics: need for special contact mechanisms
- Mechanics: wet bioactive environment.
- Motion planning: Movement without damaging delicate intestine wall
- Umbilical: useful for power, communication to/ from device

Simulating soft surface dynamics and in general the whole internal-to-body environments such as those of the colon is not widely available. Legged locomotion on soft ground has been considered [18] but to the authors knowledge there is no software suitable for simulation of internal body environments. In view of this, rigid pipes have been used to simulate the colonoscopy application. Hence in the simulations the pressure at the contact points (assuming rigid pipes) is monitored and controlled so that it does not exceed the threshold for causing rupturing of the colon wall.

As already mentioned CLAWAR's generic methodology uses four basic modules to interact via six variables (power, mechanics, data bus, analogue signals, digital signals and the environment) to integrate with each other to allowing application specific solutions to be formed. These main basic modules that are needed comprise sensors actuators, power supplies, computing hardware, software, communication devices and appropriate materials [2], [3]. In order to realise and implement the open design methodology we need to determine a good way of integrating the modules by having open protocols for the interfacing allowing the modules to be seen as black boxes. In order to do this we introduce the Common Module Block shown in Figure 4 where the input and output variables can be specified. The "input variables" simply state the input requirements of the module while the "output variables" categorise what is being output for the six interfacing space variables. When a few modules have been designed so that they match and can be integrated, it is hoped that a mature and robust methodology and set of standards can be determined that will be acceptable to the CLAWAR and wider robot communities. Having as reference the subsumption architecture which Brooks established [19], we can extend it to have the emphasis on modular components and how they can integrate together; the concept is shown more clearly in [20].

Module Title:		
INPUT	Extra Details	OUTPUT
Power:		Power:
Analogue		Analogue
Digital:		Digital:
Databus		Databus
Environment		Environment
Mechanics		Mechanics

Fig. 4. The Module Block showing the input and output variables

B. Search and rescue applications

This case study concentrates on the design of modular robotic systems used for search and rescue (SAR) missions. The field of SAR is an important application domain for robotic systems both for humanitarian and safety reasons. However, the operational environments are complex, hostile and dynamic and autonomous agents can fail in even simple tasks. Although current technology in tele–operated systems has made advancements, the produced systems are still unreliable and fail to achieve their expectations. Studies have shown that completely remotely operated vehicles are not the best solution, rather a mixture of autonomy and tele–operation should be used [21], [22]. The experience and the results gained from the use of SAR robots in World Trade Centre disaster following 11 September 2001 have given great insight into the reasons underlying the failure and rejection of robotic systems by the rescue teams [23]. Modularity was requested by the rescue teams as it is desirable to use the robots across the various sub–teams. Furthermore, modularity extends the lifetime and usability of the robotic systems into other types of missions. This can be made feasible by using available plug–n–play modules, even though they utilise in–house designs. Due to space and time limitations we will focus on urban search and rescue (USAR). It is usually used to describe SAR operations of survivors trapped in confined spaces under the debris of collapsed structures after natural or human–made disasters, such as earthquakes, landslides, or warfare and criminal activities.

The methodology described in Section II and Figure 1 is realised. The first step is to identify the task requirements of a robot in a USAR mission. These are,

- to navigate in a confined space leading to power autonomy, large power/weight ratios, compact size requirements,
- to search for survivors and important objects or clues that could lead to them, by using range sensors, data fusion

(a) World Trade Center 11/09/2001

(b) Damaged house after an earthquake

Fig. 5. USAR environments

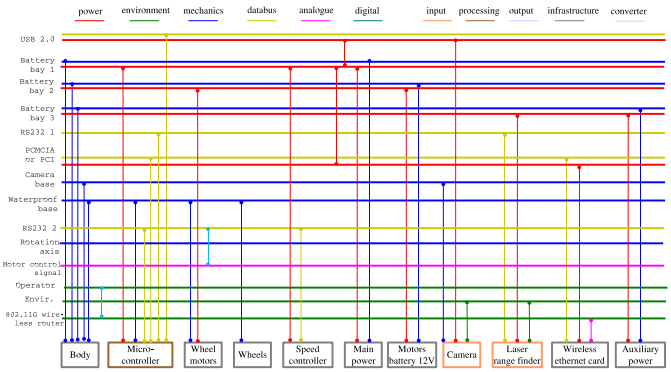

Fig. 6. CLAWAR interaction space diagram for a simple USAR robot.

techniques and autonomous search and failure recovery capabilities,
- to provide data back to the rescue base that will help in formalising a search strategy or a rescue plan,
- actively participate in the rescue operation.

The operational environments in USAR (Figure 5) can be characterised as dynamic, hostile and rough. Entry points are often narrow and difficult to reach. The terrains the robots have to navigate are unstructured and uneven, making even the simplest movements difficult without getting stuck. There is always the danger of further collapse. The light conditions are normally very poor. In addition, survivors or clues leading to them are usually covered by dust, a camouflage which makes them difficult to differentiate from the environment [23]. Even worse, due to the complete disorder of the environment the readings of individual sensor can be noisy and unreliable.

In order for the robot to achieve its mission specifications the following modules could be of use. They are grouped according to the four generic categories of modules.

- Input: camera, thermal camera and directional microphones, laser range finder, infrareds, compass, tiltmeter, gas sensors, medical sensors;
- Processing: motherboard, CPU, memory, software modules such as internal monitoring, victim recognition, mapping, localisation, hazards detection and avoidance, communication dropout recovery, etc.;
- Output: speakers, grippers, drill gun, welding gun, suction pipe, etc.;
- Infrastructure: power supply, motors, speed controllers, wireless transceiver, tether attach socket for cable communication, carrier case, water-air-toxic proof structure.

A simpler version of a possible USAR robot was chosen in order to form the interactions space diagram. Recalling that CLAWAR's interaction variables are mechanical, power, databus, analogue, digital and environment, it can be seen in Figure 6 how the modules define the interaction variables and how they interact with each other. For example the motherboard has a USB port in which a USB camera can be plugged in providing a databus and a power supply; the motors power supply is at 12V defining the power layer in which 12V devices can be plugged in.

Lastly but not least, benchmarks must be defined in order to assess the modules individually and within the integrated system. These depend on the task and user requirements as well as the operational environment. Some may have higher importance than others. We believe that for a USAR robot the benchmarks ranked from more important to least are:

- Safety
- Goal achievement-performance
- Usability
- Reliability
- Efficiency
- Technical
- Cost–expendable

Safety issues are of highest importance and a rescue team will firstly consider them in order to use the technological advancements in their missions. A robot must not compromise the safety of the rescuers and of the survivors. A robot that will set an explosion in a hazardous gas area or cause a floor to collapse due to its weight will not be accepted. For that reason part of the technical benchmarks should be considered in conjunction with the safety ones. The performance of the robot can be simply measured on how well it is able to detect survivors. For this, standard benchmarks have been defined by NIST through the RoboRescue competition [24], [25]. The usability is another important assessment criterion so that the robots can be used for other missions in the SAR operation (eg. structural integrity inspection, identification of hazardous materials, etc.). A solution to that is to have easily manually reconfigurable robots in short time. Moreover, the robot should not add to the extensive fatigue of its operator [23]. Lastly, it should provide facilitation that makes its operation and the accomplishment of the goals as easy as possible. Designs and benchmarks for that can be derived from human–machine interaction interfaces as well as from autonomous behaviours.

In order for these designs and assessments to be realised software tools are needed. As the design methodology is decomposed into system and module levels, different software tools can be used for each. A system level software tool should allow the developer to build a virtual environment similar to the real operational one, describe a robot system and its components and simulate it within the environment.

A more comprehensive list of available software tools as well as the criteria that were used to evaluate them is presented in [12]. From these results it was concluded that the "package" of Player/Stage/Gazebo (PSG) [26] is the most suitable one. Stage is a 2D simulator that allows a large number of robots to be simulated in a virtual environment. Gazebo is like Stage with the only difference that it is 3D. A drawback of Gazebo is that it is compute intensive and simulations involving large populations of robots should be avoided. Robot modules and systems can be described and used both in Stage and Gazebo. Player is a robot interface providing a network interface to a variety of robot and sensor hardware. The user can extend it with his/her own modules. The server/client architecture allows the control programs to be written in any language. These can be used for virtual robots in Stage, Gazebo as well as real ones (through TCP) with no or little modification. Lastly, PSG has interconnections with other SourceForge[1] projects such as MARIE and RobotFlow. An alternative system level simulator with similar capabilities to PSG is the commercial Webots™ [27]. However, its cost is much higher compared to the free PSG.

V. Conclusions

The design of modular robotic systems can be decomposed into system, module and protocol level designs which are expected to be carried out in parallel. Software tools are needed for these. However, the vast number of available packages makes the selection difficult and time consuming.

In this paper a design methodology and a software toolset for the design and development of robotic systems was formalised with the aid of an evaluation framework which is appropriate for this kind of selection. The toolset was formed according to the required design and assessment issues for the system, module and protocols levels of the open modularity problem. This was illustrated through two case studies focusing on search and rescue and colonoscopy applications. These show that having a formal structured approach with appropriate software toolset support, it is possible to develop and test design concepts very quickly and easily. In this way developing a modular library can be made possible so that robotics R&D can focus on the advancement of new technologies and not those that are known already. In this way the area of robotics can move forward much faster to deliver the systems that we all need.

References

[1] G. Virk, "CLAWAR: Modular robots for the future," in *Proc. of the 3rd Int. Workshop on Robot Motion and Control, RoMoCo02*. IEEE, 9–11 Nov. 2002, pp. 73–76.

[2] ——, "CLAWAR modularity for robotic systems," *The International Journal of Robotics Research*, vol. 22, no. 3–4, pp. 265–277, March–April 2003.

[3] ——, "Technical Task 1: Modularity for CLAWAR machines – specifications and possible solutions," in *2nd Int. Conference on Climbing and Walking Robots*, G. Virk, M. Randall, and D. Howard, Eds., 1999.

[4] ISO/IEC, "Standards 9126 (information technology – software product evaluation – quality characteristics and guidelines for their use)," 1991.

[5] J. Nikoukaran, J. Hlupic, and R. Paul, "Criteria for simulation software evaluation," in *Simulation Conference Proceedings*, 1998, pp. 399–406.

[6] J. Banks, "Selecting simulation software," in *Simulation Conference Proceedings*. IEEE, 8–11 Dec. 1991, pp. 15–20.

[7] L. Davis and G. Williams, "Evaluating and selecting simulation software using the analytic hierarchy process," *Integrated Manufacturing Systems*, vol. 5, no. 1, pp. 23–32, 1994.

[8] D. Eccles, "Building simulators for aerospace applications: processes, techniques, choices and pitfalls," in *Aerospace Conference Proceedings*, vol. 1. IEEE, 18–25 March 2000, pp. 517–527.

[9] X. Franch and J. Carvallo, "A quality-model-based approach for describing and evaluating software packages," in *Joint Int. Conference on Requirements Engineering Proceedings*. IEEE, 2002, pp. 104–111.

[10] N. Maiden and C. Ncube, "Acquiring COTS software selection requirements," *IEEE Software*, vol. 15, no. 2, pp. 46–56, March–April 1998.

[11] V. Hlupic and A. Mann, "Simselect: a system for simulation software selection," in *Simulation Software Proceedings*. IEEE, 1995, pp. 720–727.

[12] Y. Gatsoulis, I. Chochlidakis, and G. Virk, "Design toolset for realising robotic systems," in *Proc. of CLAWAR 2004, 7th Int. Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. Springer–Verlag, 22–24 Sep. 2004.

[13] A. Menciassi, J.H.P., S. Lee, S. Gorini, P. Dario, and J.-O. Park, "Robotic solutions and mechanisms for a semi-autonomous endoscope," in *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 2002.

[14] P. GOH and K.K., "Microrobotics in surgical practice," *British Journal of Surgery*, no. 84, pp. 2–4, 1997.

[15] H. Amano and K. T.-J. Tarn, "Development of vertically moving robot with gripping handrails for fire-fighting," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.

[16] R. Aracil and O. Reinoso, "Parallel robots for autonomous climbing along tubular structures," *Robotics and Autonomous Systems*, no. 42, pp. 125–134, 2002.

[17] L. France, A. Angelidis, P. Meseure, M.-P. Cani, F. Faure, and C. Chaillou, "A layered model of a virtual human intestine for surgery simulation," *Elsevier Science*, pp. 1–20, 2004.

[18] I. Leppanen, S. Salmi, and A. Halme, "Workpartner - hut automation's new hybrid walking machine," in *IMSRI, Clawar98*, 1998.

[19] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, Mar. 1986.

[20] I. Chochlidakis, Y. Gatsoulis, and G. Virk, "Open modular design for robotic systems," in *Proc. of CLAWAR 2004, 7th Int. Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. Springer, 22–24 Sep. 2004.

[21] K. Ali, "Multiagent telerobotics: matching systems to tasks," Ph.D. dissertation, College of Computing, Georgia Insitute of Technology, June 1999.

[22] R. Wegner and J. Anderson, "Balancing robotic teleoperation and autonomy for urban search and rescue environments," in *Proc. of Canadian Conference on AI*. Springer, 2004, pp. 16–30.

[23] J. Casper and R. Murphy, "Human–robot interactions during the robot–assisted urban search and rescue response at the World Trade Center," *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 33, no. 3, pp. 367–285, June 2003.

[24] H. Kitano and S. Tadokoro, "RoboCup Rescue: a grand challenge for multiagent and intelligent systems," *AI Magazine*, vol. 22, no. 1, pp. 39–52, Spring 2001.

[25] R. Murphy, , J. Blitch, and J. Casper, "AAAI/RoboCup–2001 urban search and rescue events: Reality and competition," *AI Magazine*, vol. 23, no. 1, pp. 27–42, 2002.

[26] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: tools for multi–robot and distributed sensor systems," in *Proc. of the Int. Conference on Advanced Robotics ICAR 2003*, 30 June – 3 July 2003, pp. 317–323.

[27] O. Michel, "Webots: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[1] http://sourceforge.net