Article:

Beccani, M, Tunc, H, Taddese, AZ et al. (4 more authors) (2015) Systematic Design of edical Capsule Robots. IEEE Design & Test, 32 (5). pp. 98-108. ISSN 2168-2356

https://doi.org/10.1109/MDAT.2015.2459591

Systematic Design of Medical Capsule Robots

Marco Beccani, Hakan Tunc, Addisu Taddese, Ekawahyu Susilo, Péter Völgyesi, Ákos Lédeczi, Pietro Valdastri

***Abstract*—Over the last decade, researchers have started exploring the design space of Medical Capsule Robots (MCRs): devices that operate autonomously within the human body and can monitor, diagnose, prevent, and cure diseases. MCRs are severely resource constrained devices in size, power and computational capacity. As such, the design process for MCRs is time consuming and it requires deep expertise in multiple domains. To open up the field and unlock the vast clinical potential of these devices as diagnostic and interventional tools, we have created an open source platform consisting of a library of modular hardware and software components and a web-based collaborative design environment.**

I. Introduction

The increasing availability of miniaturized components at rapidly decreasing prices is enabling exploration of new application fields in ways that researchers were only dreaming about a decade ago. In the medical field, miniaturization of electronics has enabled swallowable endoscopic capsules to acquire pictures of the small intestine—a region extremely difficult to reach otherwise as it is located in the middle of the alimentary canal—with minimal disruption to the patient [1]. Unfortunately, this technology has not successfully expanded to other anatomical districts, such as the stomach or the colon, where cancer strikes much more severely. Colorectal cancer alone affects more than 170,000 Americans each year and kills approximately 50,000, justifying over 15 million screening colonoscopies per year in the US. Functions such as active locomotion, advanced diagnostics, biopsy collection, cancer removal, drug delivery or tissue treatment would enable smart capsules, referred to as Medical Capsule Robots (MCRs) in this paper, to eventually replace flexible endoscopy, thus dramatically improving healthcare delivery with a tremendous impact on the life of millions of people worldwide.

Progress in the field of MCRs, however, requires advanced skills in embedded systems, miniaturized electronics, and packaging as a prerequisite before even starting to address the challenge of implementing one of the specific functions listed above. To lower the barrier to design space exploration and to shorten the time-to-prototype in the field of MCRs, we propose an open source platform for rapid design and development of capsule robots. Inspired by Lego's modularity, our platform includes libraries of hardware and software components, a flexible backbone for plug-n-play connectivity of the hardware modules, and a web-based design environment where developers can implement, program, and verify a specific design before assembling a prototype. Thanks to the open source approach, users can extend the repository with their own modules thus establishing channels of collaboration for design reuse. Similar paradigm shifts are currently taking place in robotics since the introduction of ROS [2], and in surgical robotics with the open source software libraries for the Intuitive Surgical Da Vinci robot [3] and the RAVEN II modular hardware platform [4]. Looking forward, our goal is to open the field of capsule robot design to a wider community and, at the same time, create better designs through advanced tool support. Furthermore, our approach can be extended to adjacent segments of medical technology, such as wearable monitoring systems and implantable healthcare devices.

M. Beccani, A. Taddese, E. Susilo, and P. Valdastri are with the STORM Lab, Department of Mechanical Engineering, Vanderbilt University, Nashville, TN 37235.
Email: p.valdastri@vanderbilt.edu
A. Taddese, H. Tunc, P. Völgyesi and A. Lédeczi, are with the Institute for Software Integrated Systems, Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37212

II. Background

Research in MCRs systems from recent progress in Wireless Capsule Endoscopy (WCE). The first clinical case of WCE was reported in 2000 [1]. The platform was commercialized one year later by Given Imaging, Ltd. Since then, more than 1,500,000 endoscopic capsules have been used worldwide. An endoscopic capsule consists of an external biocompatible shell, typically the size of a large antibiotic pill (11 mm in diameter, 26 mm in length), a camera, a control and communication unit, and an energy source. Antennas are placed on the patient to receive wireless data, and the camera images are stored on the device for the doctor to review upon completion of the procedure.

WCE has become the gold standard for the diagnosis of diseases in the small intestine where biopsies and active locomotion are of little importance. As for the large intestine, in February 2014, the Food and Drug Administration approved the Given Imaging PillCam Colon for incomplete colonoscopies [13] (i.e. if the endoscopist fails to visualize the entire colon via standard colonoscopy, the patient can be referred to WCE), but not for colorectal cancer screening. During a colonoscopy, it is ideal for the endoscopist to be able to maneuver the camera view in order to further explore suspicious lesions. Current capsules lack this capability since they only move by peristalsis. In addition, capsules also lack the ability to interact with the gastrointestinal tissue. In order to replace colonoscopy in colorectal cancer screening, future capsules must be able to move under the control of an external operator and be able collect biopsy samples and clip polyps.

To address these limitations, researchers are currently pursuing a broad range of innovative MCR designs. While a selection of significant MCRs is presented in Fig. 1 and described in this section, the interested reader can refer to a number of recent review papers [14]–[16] for a broader overview of the field.

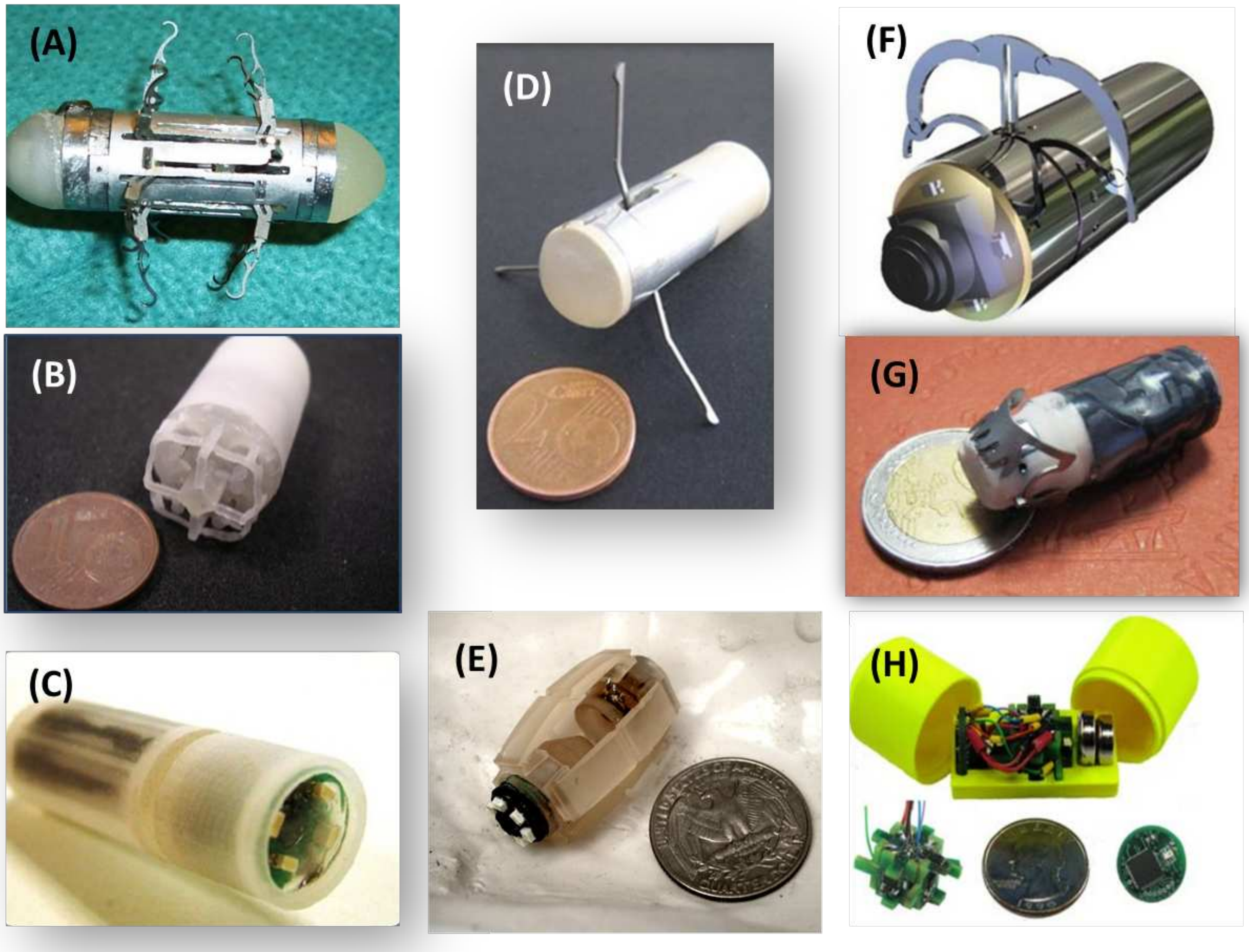

Fig. 1: Selection of MCRs developed to date. Diameter and length of each prototype are provided in mm in brackets: (A) 12-leg capsule [5] (11 × 30). (B) Submarine-like capsule [6] (11 × 29). (C) Magnetic capsule [7] (11 × 29). (D) Legged-magnetic hybrid capsule [8] (14 × 38). (E) Magnetically actuated capsule endoscope (MASCE) [9] (15 × 40). (F) Drug delivery capsule [10] (11 × 25). (G) Surgical clip releasing capsule [11] (11 × 30) (H) Magnetic capsule with localization [12] (11 × 26).

To date, the most active field of research in MCR has been active locomotion. Legs can propel a camera pill forward and backward inside the colon, while at the same time enlarging the lumen to provide adequate visualization. The capsule presented in Fig. 1.A includes two Direct Current (DC) brushless motors, magnetic encoders, a vision system, and a wireless microcontroller to crawl forward and backward inside the colon [5]. In order to adapt MCRs to perform upper gastrointestinal endoscopy, the stomach can be filled with water to provide a large open space ideal for inspection by a submarine-like capsule as the one presented in Fig. 1.B. This capsule uses four independent propellers actuated by embedded DC brushed motors, and its speed and direction can be controlled via wireless communication with a joystick [6]. In this case, the signal coming from an embedded accelerometer can be used to balance the capsule thrust direction by controlling (via pulse width modulation) the rotational speed of each of the four motors.

The legged and submarine-like capsules both have actuators on board. This increases the complexity of the MCRs and requires space on board for the actuators and the power source to operate them. An alternative solution that addresses these limitations consists in magnetic coupling between a permanent magnet inside the MCR and a controllable source of magnetic field external to the patient. MCRs that leverage magnetic coupling for active manipulation are presented in Figs. 1.C-D.

The MCR in Fig. 1.C contains a permanent magnet, magnetic field sensors, and a triaxial accelerometer to enable position and orientation detection via the use of a computationally efficient magnetic field model [7], [17]. A wireless microcontroller embedded onboard acquires data from the sensors and sends them to the external unit, which computes the necessary external magnet motion to achieve the desired internal capsule motion. The main disadvantage of pure magnetic locomotion is the lack of ability to distend tissue away from the capsule in order to enable the capsule to move or to improve visualization of the entire intestine surface. To address this, a hybrid legged-magnetic approach was pursued in [8], where the MCR had a single set of actuated legs, along with simultaneous external magnetic locomotion, as shown in Fig. 1.D. Limiting the use of the onboard mechanism to situations where external locomotion is not effective reduces the power consumed by the capsule, extending battery lifetime. A similar hybrid solution was proposed in [9], where magnetic coupling worked in synergy with the compliant body of the MCR to achieve locomotion and localized drug delivery. This capsule, presented in Fig. 1.E, can be used to deliver bacteria like micro-propelled robotic swarms [18], which can be used to perform massively parallel tasks and access smaller spaces that are out of reach for larger MCRs. A different approach to drug delivery was proposed in [10], where an onboard brushless DC motor was used to stabilize the capsule and deploy the drug (Fig. 1.F). The anchoring mechanism can resist peristaltic pressure through the deployment of an integrated holding mechanism and targeted drug delivery is possible through the activation of a needle which can be positioned in a 360° envelope.

The first capsule that can carry out a surgical intervention was proposed in [11] for controlled release of a surgical clip that can stop localized bleeding in the colon. This MCR (Fig. 1.G) uses embedded permanent magnets, enabling external locomotion via magnetic fields, and a pre-loaded clip, which can be fired by the action of a miniature integrated DC brushless motor, based on a wireless command.

Besides active locomotion and tissue interaction, MCR localization is a relevant research stream in the field. The position of the MCR can be used for diagnostic purposes (i.e. to associate a position inside the gastrointestinal tract to a lesion identified by the capsule) or for controlled manipulation (i.e. if the MCR position is known, more effective locomotion strategies can be implemented). An example of the latter is the MCR in Fig. 1.H, which integrates magnetic field sensors and a wireless microcontroller to inform magnetic manipulation obtained via an externally controlled source [12].

The vast majority of the MCR developed thus far, including the ones in Fig. 1, share the same crosscutting constraints such as (1) size—ideally, a capsule device should be small enough to swallow or to enter natural orifices without requiring a dedicated incision; (2) power consumption—given the limited space available on board, energy is limited; (3) communication bandwidth—wireless signals must be transmitted through the human body with sufficient data rate; (4) fail-safe operation—since the device is deep inside the human body, medical personnel has no direct access to it; and (5) effective interaction with the target site according to the specific functions the device is required to fulfill.

Given these constraints, it is possible to abstract a general system architecture for an MCR, consisting of (A) a Central Processing Unit (CPU) that can be programmed by the developer to accomplish a specific task; (B) a communication submodule that links the device with user intent; (C) a source of energy that powers the system; and (D) sensors and (E) actuators, both of which interact with the surrounding environment to accomplish one or more specific tasks. It is also desirable for the designers to have a model of the environment in order to predict the effectiveness of the design in accomplishing a specific task.

III. Approach

Based on the identified crosscutting constraints, we have developed a modular hardware with specific components of the generic MCR architecture that can be added, removed, customized and interchanged as needed. On the software side—building on a low-footprint execution framework—we provide a component-based MCR library using strong encapsulation, well-defined interfaces and scalable integration mechanisms, all supported by a model-based design environment.

To support the research community in adopting and using the integrated design environment, all the material developed by the authors is open source and available online at `pillforge.github.io`. This repository contains all the modules presented in this manuscript, including their Bill of Materials (BOM), schematics, fabrication files, Computer Aided Design (CAD) models, as well as code examples and the documentation on how to use the proposed environment. The public code repository also contains the sources of the complete web-based design environment. In addition, the repository contains the source code of the design environment with a description of how to run and use it locally. For the ease of use, we also provided the Docker container-based deployment of the project.

IV. Hardware Overview

Each of the hardware modules developed to date targets one of the five major subsystems of MCRs, i.e. computation, wireless communication, power management, sensing and actuation. A list of the developed hardware modules and their capabilities is provided in Table I. A versatile communication and power distribution backbone is provided by a flexible circuit board with multiple connectors for greatly reducing the time needed to assemble MCR prototypes. Beyond our continuously expanding set of modules, we encourage novel contributions from the wider research community by publishing relatively simple design rules and interface specifications on our web-based repository.

A. Computation

The CPU handles local computation, sensor management and other housekeeping tasks of the MCR. It must be small enough in size to be embedded into the capsule, and must operate in low-power modes to maximize the lifetime of the device, with built-in support for directly interfacing with various sensors, actuators and wireless transceivers.

Our initial CPU module is based on the fifth generation MSP430 from Texas Instruments. The developed miniaturized board has a diameter of 10.6 mm and a thickness of 1.6 mm. A dedicated Serial Peripheral Interface (SPI) bus—available through a miniature connector on top of the CPU module—is used with the wireless modules. This connector enables

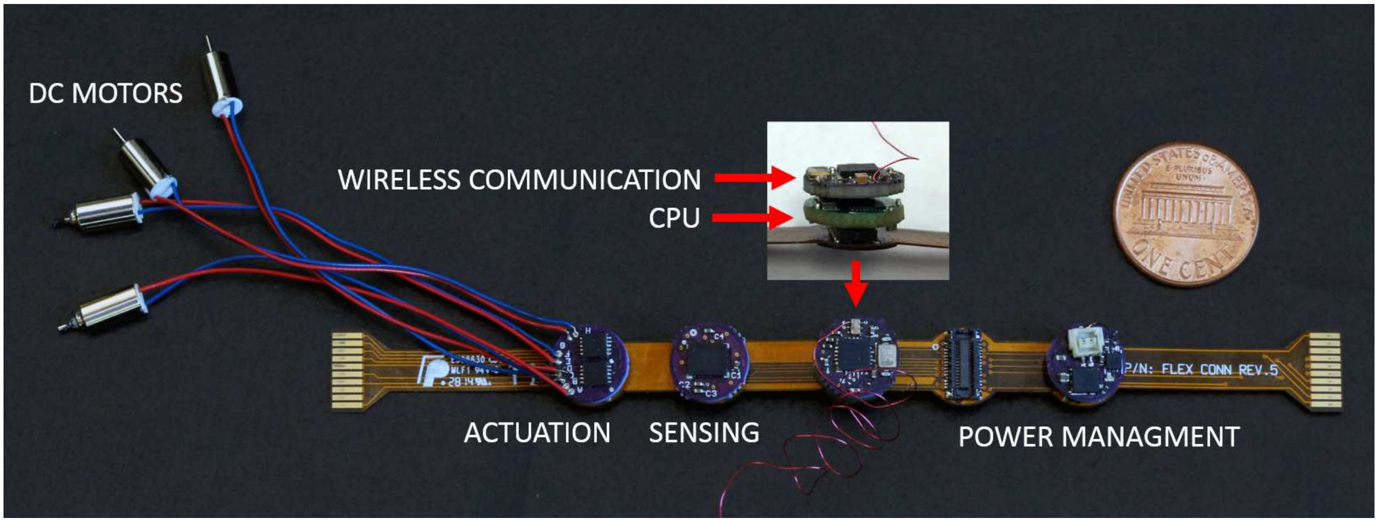

Fig. 2: The flexible circuit with hardware modules before folding in a shape that can be integrated inside an MCR. The flexible circuit can host up to five different modules. In this case the wireless communication module is plugged on the CPU module in the central slot. The other slots are respectively hosting an actuation module, a sensing module and the power managing module.

developers to choose from alternative radio technologies and frequency bands according to the needs of the application. Additional SPI and Inter-Integrated Circuit (I^2C) buses on the flexible circuit are used to interface with sensor and actuator modules.

B. Wireless Communication Modules

Many MCR applications require a wireless communication channel for transmitting information from inside the human body. This communication must often be performed with relatively high data rates for real-time applications with transmission power bound to medically safe limits. Sub 1 GHz carriers with lower energy absorption rates in human tissues are the most suitable for medical devices [19]. However, as shown in [19], a 2.4 GHz carrier, which is used in ZigBee and Bluetooth applications, can be adopted for intra-body data communication as well. Our platform supports both bands and could enable data communication between smart phones or tablets and the MCR.

C. Power Management Module

The power management module provides regulated power from the battery to the rest of the hardware modules. This module integrates an Low Drop-Out (LDO) to supply a single fixed 3.3 V power rail. It can also charge and monitor the state of the battery.

D. Sensing Modules

Sensing modules implemented thus far consist of commercially available small package digital or analog sensors such as accelerometers, gyroscopes, pressure sensors and Hall-effect sensors. In general, any sensor with a digital interface (SPI or I^2C) operating at a maximum voltage of 3.3 V can be physically connected to the microcontroller (MCU) and be accessed by the software. Similarly, analog sensors can be connected to analog to digital converter (ADC) channels of the MCU or to external ADC devices with digital interfaces to the MCU.

E. Actuation Modules

Actuation Modules provide multiple digital interfaces (e.g., Pulse Width Modulation (PWM), General Purpose I/O (GPIO)) for utilizing both brushed (BDCC) or brushless (BSLDCC) DC motor controllers. These modules have been designed to enable internal active locomotion or to actuate internal mechanisms of capsules endoscope such as the ones presented in Section II.

F. The Flexible Circuit Backbone and the Base Station

Connectivity between the modules is achieved with a flexible circuit on which modules are mounted before being folded to form the body of the MCR. The flexible circuit has a width of 10.4 mm and a length of 72.5 mm and can be folded down to a 14 mm length when all slots are populated. The backbone can host up to five different modules using 30-pin miniature connectors. The central connector is dedicated for the CPU module, while the other four slots accept arbitrary modules, thus providing extreme flexibility to the designer. Data connectivity is achieved by sharing SPI, I^2C, ADC inputs, and GPIOs with the CPU. To prevent damages caused by incorrect mounting of the modules, the flexible circuit has been designed with power and ground connections mirrored on each side of the connector. The platform comes with a base station that is used to exchange data wirelessly between the MCR and the user interface. In addition, it mirrors the architecture of the flexible circuit making it possible to program, test and debug all modules before their integration into a miniature device. The documentation related to the base station is available in the online repository.

Module Name	Functionality	Integrated Circuit (IC)	Diameter (mm)	Thickness (mm)	Current Max. (mA)
CPU	Processing / Computation	MSP430F5528	10.5	3.84	2.32
433 MHz Transceiver	Wireless Communication	CC1101	10.5	3.94	29.2
3DA	3D Accelerometer	LIS331DLH	10	4.04	0.25
3DG	3D Gyroscope	L3GD20	10	4.04	6.1
3DM	3D Magnetometer	LIS3MDL	10	4.04	0.29
6DAG	3D Accel. Gyro.	LIS330DLC	10	4.04	0.01
6DAM	3D Accel. Magnet.	LIS330D	10	4.04	0.3
9DAMG	3D Accel. Gyro. Magnet.	LSM9DS0	10	4.04	6.15
PT	Pressure and Temperature	MPL115A1	10	4.25	0.005
2AF&ADC	2 Ch. Front End ADC	AD623 & ADS8320	10	4.35	2.57
8CHADC	8 Ch. ADC	AD7689	10	3.94	3.78
BDCC	Actuation	(2 ×) A3901	10	3.69	4.8
BSLDCC	Actuation	BH67172NUX	10	3.69	4.5
PM	Power Management	NCP606, LTC4065, LTC2942	10	3.84	500 (Max)

TABLE I: Summary of the hardware modules currently available.

V. Software Design Environment

Current software development practices for MCRs are based on low-level languages (C and assembly) to implement bare metal applications. These typically run directly on top of the hardware with no operating system support. Debugging and fine-tuning hand-crafted applications is time consuming, error-prone, and requires deep expertise with limited opportunities for effective code reuse. Just as a component-based approach on the hardware side is beneficial, it alleviates many of these problems on the software side as well. TinyOS [20] is a configurable modular operating system that was specifically designed for such resource constrained platforms as Wireless Sensor Networks (WSN). Since MCRs share many of their characteristics with WSNs, such as power, size and other severe resource constraints; TinyOS is a natural choice for this domain as well. TinyOS has a notion of the software component, which is an encapsulation of software functionality that exposes well-defined interfaces to other components. Applications in TinyOS are composed of components which are "wired" together with other components through their interfaces. In addition to TinyOS core components, users can define other application-specific or re-usable middleware components. Hardware components are also supported by driver-level TinyOS components. Another key property of TinyOS is that only the components that are actually used in the application become part of the executable code.

The goal of our project is to open up the area of MCRs to people who are not embedded hardware and software experts. The component-based approach on both the hardware and software sides goes a long way toward that goal. However, we believe that the current level of abstraction—i.e. the TinyOS component model—can and should be raised to reach our goal. We advocate a high-level—primarily visual—design environment similar to Simulink or Labview. An intuitive graphical domain-specific representation, corresponding user interface and automatic code generation significantly lowers the barrier of entry into MCRs.

However, creating sophisticated and user friendly design environments from scratch is a costly and long endeavor. Instead, we applied a configurable graphical environment called Web-Based Generic Modeling Environment (WebGME) [21]. WebGME supports collaborative modeling similar to Google Docs and the models are stored in a version controlled database in the cloud. For the MCR domain, we defined a modeling language that captures both the hardware and software components as well as their interconnection. This language allows users to create TinyOS software components within the visual environment. Furthermore, since the TinyOS code-base has been automatically imported, the users can browse existing components as well and model the entire application. We developed a custom visualization module that shows the graphical view of the modular TinyOS application side-by-side with the code view, as presented in Fig. 3. A change in either view is reflected in the other. The TinyOS development tools run in the cloud, so there is no need to install the toolchain on the user's computer. All that is needed is a browser and a network connection.

VI. An Application Example: the Propeller-based Swimming Capsule

While most of the MCR developed to date can be replicated by combining the existing modules or by creating new ones, in this section we provide a concrete example by redesigning a submarine-like capsule with the proposed design environment. This MCR is similar in functionality to the custom-built capsule in [6] and embeds 6DAG, BDCC, CPU, and wireless communication modules (see Table I), 4 DC motors and a 50 mAh battery. The shell of the capsule and the propellers are made by rapid prototyping. The assembled MCR resulted in an overall length of 42 mm and a diameter of 12.5 mm. Fig. 4 shows the assembled prototype.

Fig. 3 shows the component visualization and source code views of the TinyOS application used in the capsule. Boxes in the visual editor represent TinyOS components, which can themselves be composed of other components. Users can navigate through the hierarchical structure of component compositions by double clicking into the boxes. A blue arrow in the figure designates a "wiring", which is a TinyOS language construct used to map a function call inside one component to a function definition in another component. Custom components can be created using the code editor, which simplifies component creation by allowing the user to focus on essential parts of the code while automatically generating all other boilerplate code.

This application acquires accelerometer and gyroscope measurements from the LSM330DLC component using the stan-

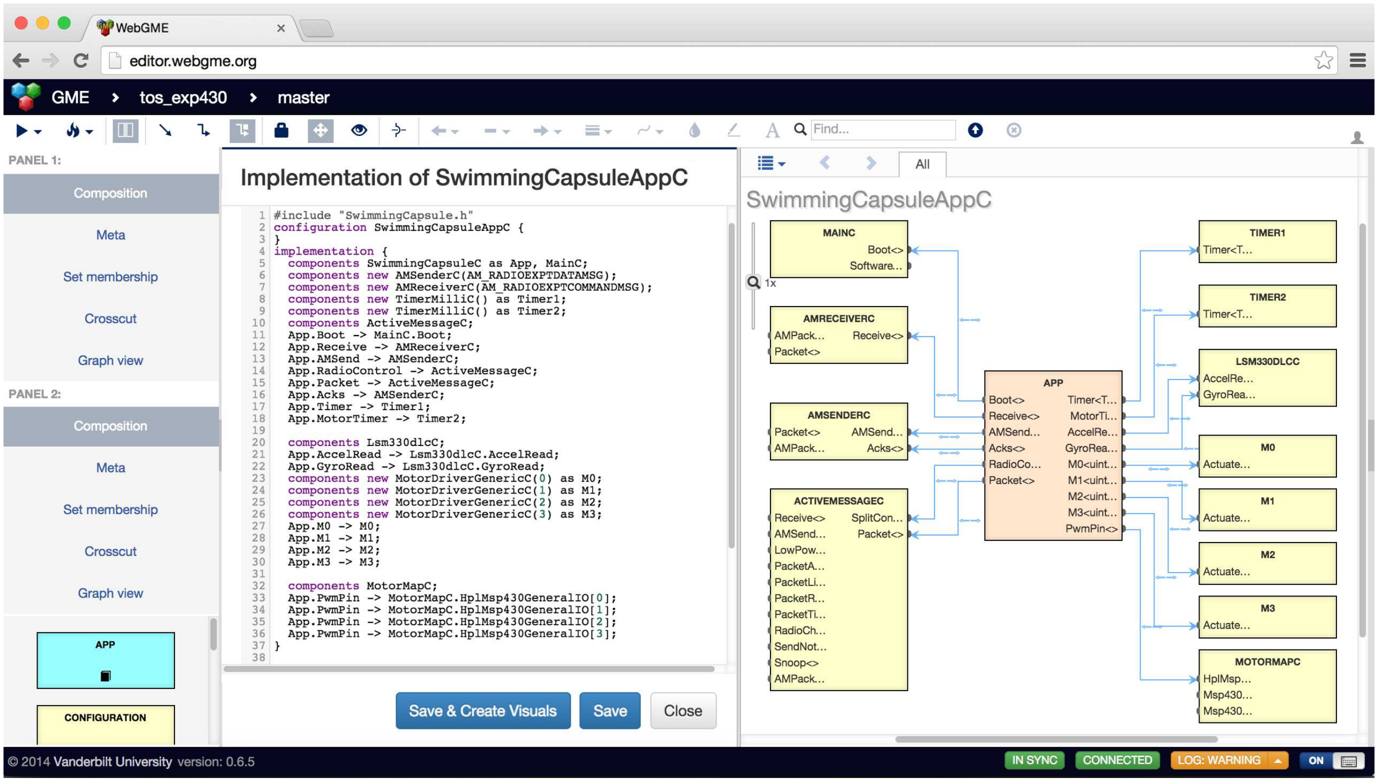

Fig. 3: Code View and Software Visual Editor of the MCR Design Environment.

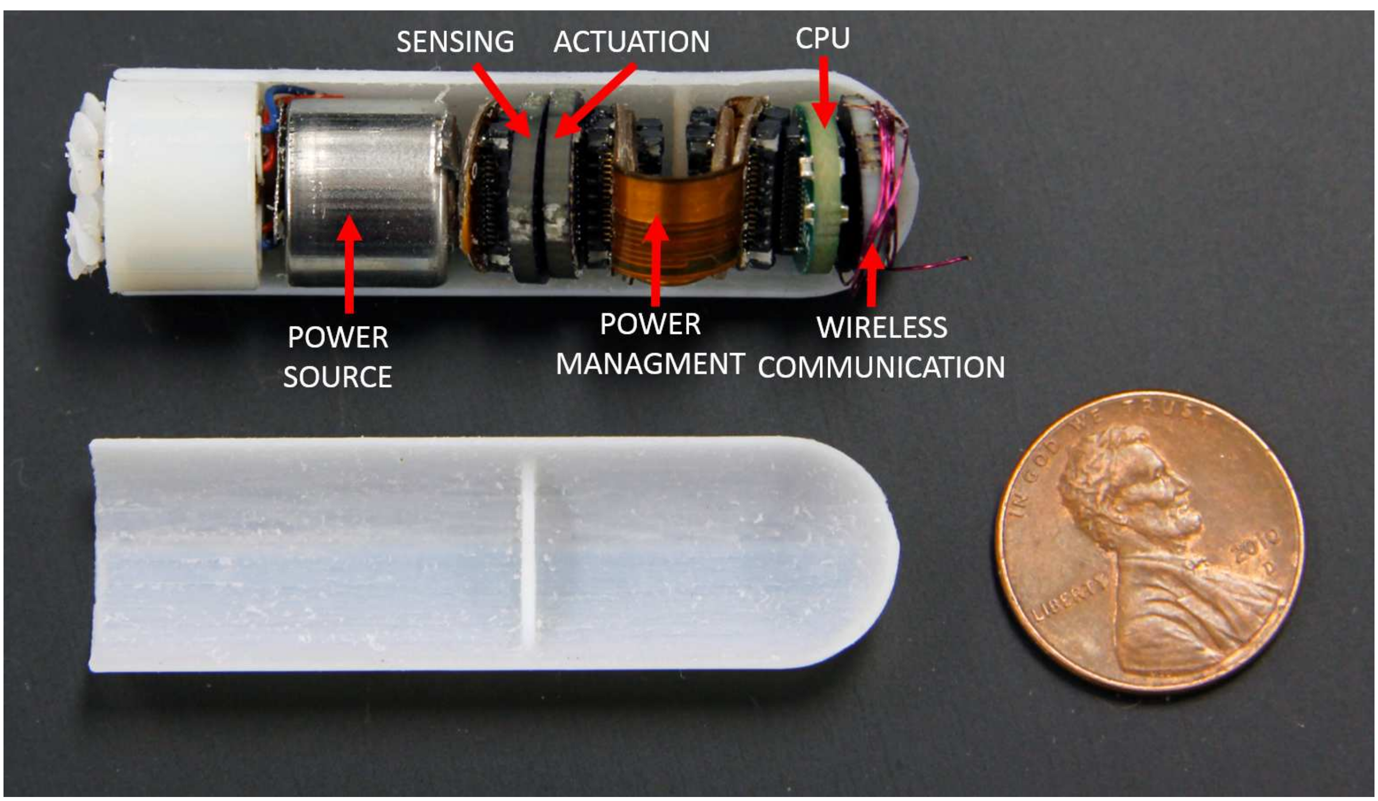

Fig. 4: The swimming capsule without the lateral shell to show the internal hardware components.

dard *Read* interface at a sampling rate set using the Timer1 component. Sensor data is sent using standard TinyOS interfaces (*AMSend* and *Receive*) to the base station, which responds with an actuation command. Wireless communications between the base station and the miniature MCR is achieved with the 433 MHz transceiver module. Based on the content of the actuation command, the application drives the four motors with varying duty cycles using the *Actuate* interface exposed by each motor component. The Timer2 component is used to turn off the motors after the period of time specified in the actuation command.

VII. Future Work

Regarding the hardware, future work will aim to extend the library to include additional microprocessors, sensor and actuator modules, and wireless transceivers implementing different communication protocols and carrier frequencies (e.g. Low Power Bluetooth). In particular, the authors are exploring to extend the proposed platform to ARM Cortex microprocessors (e.g. the CC2650 from Texas Instruments). This module consists of a 2.4 GHz radio and a high performance ARM CPU, which will further reduce the folded capsule size. Other devices that integrate similar CPUs and other wireless communication protocols will be made available in the future.

On the software side, the application of TinyOS and the corresponding web-based design environment was a significant step towards developing reusable software components and composing embedded applications on resource constrained hardware platforms. However, the scope of the visual representation is limited to the static structure of software artifacts: *temporal behavior, energy and resource consumption and hardware dependencies* are hidden or implicit in the model. Providing analysis, validation and verification tools in these domains require more detailed, multifaceted models to be built. However, continuous refinement of the visual language is one of the strengths of the underlying configurable framework.

As part of our ongoing efforts, we are developing a state machine-based representation for the module components where designers can easily reason about potential application states and valid transitions with corresponding triggering events. This feature will help the designers to develop and modify their applications rapidly and will provide a high-level view of the application's logic in one place. Currently, the same information is hidden across several functions and is spread over multiple source files. Moreover, we are developing a higher level modeling abstraction, where instead of building from fine-grained TinyOS components, an MCR application is created by selecting and configuring a handful of composite *features*, such as wireless communication, imaging, self-propelled movement, or self-localization. Such larger design partitions include pre-built templates of TinyOS components, optional surrogate simulation models, hardware dependencies (concrete hardware modules and their placement on the flexible circuit board), power profile and application-level interfaces. Our goal with the higher abstraction level and coarse-grained modules is to provide a truly intuitive and rapid design-space exploration tool with analysis, simulation and synthesis capabilities. Note that, in contrast with the lower-level more generic TinyOS component model, these modeling abstractions will target MCRs and our hardware ecosystem specifically.

VIII. Conclusions

In this work, we have introduced the first steps toward an open architecture and supporting design environment for MCR. The goal is to speed up the creation of the next generation of MCRs by leveraging an extensible library of hardware and software modules using an interactive web-based graphical environment. The designer - whether an expert or a novice in embedded systems - will be able to focus on selecting pre-designed modules and programming the MCR to fulfill a specific task, rather than designing, fabricating, and connecting together miniature electronic boards. This will speed up the time-to-prototype, reducing the gap from concept to experimentation.

The main downside of a modular approach, however, is that a system made of modules is not optimized for the particular application, and it usually causes overall an increment of the system size if compared to a custom device. The best course of action for MCR designers would therefore be to validate their hypotheses and preliminary designs using the proposed platform, and then move to a custom approach as soon as they are satisfied with the results. At that stage, compliance with regulatory standards and guidelines (e.g. IEC 60601) must also be addressed. While the current modules available in the on-line repository should be compliant with the IEC 60601 standard, a rigorous risk analysis and compliance check were both outside the scope of our work.

The proposed approach for rapid prototyping of capsule robots follows the ”democratization” of embedded systems promoted by the introduction of Arduino and other open source platforms, and lends itself to adoption well beyond medical research. In particular, we plan to release a simplified version of our architecture that can be adopted by pre-college students and by the growing number of tech enthusiasts from the maker movement. While it is too soon to speculate on the broader implications of the adoption of our approach, we hope that lowering the barriers to the design of new medical capsule robots will lead to a day when colorectal cancer screening will be as easy as swallowing a capsule.

IX. Acknowledgment

This work was supported by the National Science Foundation under Grants CNS-1239355, IIS-1453129, IIP-1506285, and 1445197. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] G. Iddan, G. Meron, A. Glukhovsky, and P. Swain, “Wireless capsule endoscopy,” *Nature*, vol. 405, no. 6785, pp. 417–418, 2000.
[2] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.

[3] Z. Zhang, A. Munawar, and G. S. Fischer, "Implementation of a motion planning framework for the davinci surgical system research kit," in *The Hamlyn Symposium on Medical Robotics*, 2014, p. 43.
[4] B. Hannaford, J. Rosen, D. W. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. N. Kosari, and L. White, "Raven-ii: an open platform for surgical robotics research," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 4, pp. 954–959, 2013.
[5] P. Valdastri, R. Webster, C. Quaglia, M. Quirini, A. Menciassi, and P. Dario, "A new mechanism for mesoscale legged locomotion in compliant tubular environments," *Robotics, IEEE Transactions on*, vol. 25, no. 5, pp. 1047–1057, 2009.
[6] G. Tortora, P. Valdastri, E. Susilo, A. Menciassi, P. Dario, F. Rieber, and M. O. Schurr, "Propeller-based wireless device for active capsular endoscopy in the gastric district," *Minimally invasive therapy & allied technologies: MITAT : official journal of the Society for Minimally Invasive Therapy*, vol. 18, no. 5, pp. 280–90, 2009.
[7] G. Ciuti, P. Valdastri, A. Menciassi, and P. Dario, "Robotic magnetic steering and locomotion of capsule endoscope for diagnostic and surgical endoluminal procedures," *Robotica*, vol. 28, pp. 199–207, 3 2010.
[8] M. Simi, P. Valdastri, C. Quaglia, A. Menciassi, and P. Dario, "Design, fabrication, and testing of a capsule with hybrid locomotion for gastrointestinal tract exploration," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 2, pp. 170–180, 2010.
[9] S. Yim and M. Sitti, "Shape-programmable soft capsule robots for semi-implantable drug delivery," *Robotics, IEEE Transactions on*, vol. 28, no. 5, pp. 1198–1202, Oct 2012.
[10] S. P. Woods and T. G. Constandinou, "Wireless capsule endoscope for targeted drug delivery: mechanics and design considerations," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 4, pp. 945–953, 2013.
[11] P. Valdastri, C. Quaglia, E. Susilo, A. Menciassi, P. Dario, C. Ho, G. Anhoeck, and M. Schurr, "Wireless therapeutic endoscopic capsule: in vivo experiment," *Endoscopy*, vol. 40, no. 12, p. 979, 2008.
[12] K. Miller, A. Mahoney, T. Schmid, and J. Abbott, "Proprioceptive magnetic-field sensing for closed-loop control of magnetic capsule endoscopes," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 1994–1999.
[13] "Given imaging," 2015. [Online]. Available: http://www.givenimaging.com
[14] P. Valdastri, M. Simi, and R. J. Webster III, "Advanced technologies for gastrointestinal endoscopy," *Annual Review of Biomedical Engineering*, vol. 14, no. 5, pp. 397–429, 2012.
[15] L. J. Sliker and G. Ciuti, "Flexible and capsule endoscopy for screening, diagnosis and treatment," *Expert Rev Med Devices*, vol. 11, no. 6, pp. 649–666, 2014.
[16] P. Slawinski, K. L. Obstein, and P. Valdastri, "Emerging issues and future developments in capsule endoscopy," *Techniques in Gastrointestinal Endoscopy*, 2015, in press, available on-line.
[17] C. Di Natali, M. Beccani, and P. Valdastri, "Real-time pose detection for magnetic medical devices," *Magnetics, IEEE Transactions on*, vol. 49, no. 7, pp. 3524–3527, 2013.
[18] G. Wei, P. Bogdan, and R. Marculescu, "Efficient modeling and simulation of bacteria-based nanonetworks with bnsim," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 12, pp. 868–878, 2013.
[19] P. Valdastri, A. Menciassi, A. Arena, C. Caccamo, and P. Dario, "An implantable telemetry platform system for in vivo monitoring of physiological parameters," *IEEE Trans. Inf. Technol. Biomed.*, vol. 8, no. 3, pp. 271–8, 2004.
[20] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer *et al.*, "Tinyos: An operating system for sensor networks," in *Ambient intelligence*. Springer, 2005, pp. 115–148.
[21] M. Maróti, T. Kecskés, R. Kereskényi, B. Broll, P. Völgyesi, L. Jurácz, T. Levendoszky, and Á. Lédeczi, "Next generation (meta) modeling: Web-and cloud-based collaborative tool infrastructure," in *8th International Workshop on Multi-Paradigm Modeling MPM 2014*, 2014, p. 41.

Marco Beccani (S'11) received a Master's degree in Electronic Engineering from the University of Pisa, Pisa, Italy, in 2010. Since 2011 he is a PhD student in Mechanical Engineering at Vanderbilt University. He is a member of the Science and Technology Of Robotics in Medicine (STORM) Lab, and his field of research is miniaturized embedded system design for wireless robotic capsular endoscopy and robotic surgery.

Hakan Tunc is a Ph.D. student and a research assistant at the Institute for Software Integrated Systems, Vanderbilt University. His research focuses on design environment for miniature medical robotics.

Addisu Taddese Addisu Z. Taddese is a Ph.D. student at Vanderbilt University. His research focuses on miniature medical robotics and their application in surgical procedures. He is an NSF graduate fellow and student member of IEEE and ACM.

Ekawahyu Susilo Ekawahyu Susilo is a Postdoctoral Scholar in Mechanical Engineering, Vanderbilt University. His major is in wireless embedded system design and miniaturization for low-power applications. He received his PhD degree, with honor, from Scuola Superiore Sant'Anna (SSSA), Pisa, Italy in 2009. He joined Decawave Ltd. in 2010, developing RTLS algorithm using IEEE 802.15.4a UWB PHY/MAC compliant devices with 10 cm precision in three dimensional coordinates. Currently, he is joining STORM Lab at Vanderbilt University.

Peter Volgyesi is a Research Scientist at the Institute for Software Integrated Systems at Vanderbilt University. He is one of the architects of the Generic Modeling Environment, a widely used metaprogrammable visual modeling tool, and WebGME - its modern web-based variant. As PI on two NSF funded projects Mr. Volgyesi and his team recently developed a low-power software-defined radio platform (MarmotE) and a component-based development toolchain targeting multicore SoC architectures for wireless cyber-physical systems.

Ákos Lédeczi is an Associate Professor of Computer Engineering and a Senior Research Scientist at the Institute for Software Integrated Systems at Vanderbilt University. He has an M.Sc. from the Technical University of Budapest and a Ph.D. from Vanderbilt, both in Electrical Engineering. His research interests include wireless sensor networks and model integrated computing.

Pietro Valdastri (M'05, SM'13) received a Ph.D. degree in Biomedical Engineering from Scuola Superiore Sant'Anna (SSSA), Italy. He is an Assistant Professor of Mechanical Engineering at Vanderbilt University and the Director of the STORM Lab. Dr. Valdastri is actively involved in robotic endoscopy and surgery, design of magnetic mechanisms, and design of capsule robots for medical applications.