A Cyber-Infrastructure for Supporting K-12 Engineering Education thorough Robotics

William M. Mongan and William C. Regli
Drexel University
Philadelphia, PA USA
{wmm24, regli}@cs.drexel.edu

Abstract

The Cyberinfrastructure Training, Education, Advancement, and Mentoring for Our 21st Century Workforce (CI-TEAM) project created an online repository of multi-disciplinary engineering modules, with an aim to enhance engineering informatics education at the undergraduate/graduate level as well as to inspire and motivate K-12 students towards engineering disciplines. Using a Wiki as a central repository facilitated the representation and storage of these engineering artifacts for easy and universal access. Particularly, because the concept of a Wiki is well understood by students at various age levels and discipline, and because the Wiki is ubiquitous, students can access and even contribute to the modules and data available there. These modules are made available with the hope that students with negative impressions about engineering disciplines can access this information and become excited about the challenges and possibilities contained therein.

Introduction

The Cyberinfrastructure Training, Education, Advancement, and Mentoring for Our 21st Century Workforce (CI-TEAM) project[1] operated under the NSF Cyber-Infrastructure Program, with an aim to create a comprehensive, multi-disciplinary approach to teaching engineering modeling. The team created novel ways in which to train future generations of engineering and computer science students to build physically realized systems for important applications in medicine, civil engineering, search and rescue, and homeland security. This effort yielded a number of beneifts for teaching engineering informatics (the science of representation, simulation, archiving, and reuse of engineering knowledge in transformative ways) as well as engineering through robotics; additionally, this work produced a repository of bio-inspired robot designs for K-12 education.

The numerous Cyber-Infrastructure reports from the NSF and NRC over the past five years have all lamented the fact that the type of inter-disciplinary engineer so desperately needed simply does not exist in adequate numbers. There are not enough educational initiatives and programs to produce these new engineers, nor have the standard, stove-piped curriculas of engineering and computer science departments adapted to this need. Numerous reports from the National Academies and elsewhere echo these concerns, extensively documenting how these disciplinary boundaries continue to impede the innovation process by delaying the conversion of new ideas into innovative products (Piasecki *et al.* 2004; LTE 2003; NCAR 2003; SIN 2003). Negative or incorrect perceptions about science and engineering careers held by adolescents (and their parents) exacerbate these problems.

To address this, we developed a repository of snake-based and other bio-inspired robots, and have made them accessible for inspiring, motivating, and teaching students both engineering concepts as well as core curricular subjects with a practical grounding in engineering. This education begins at the K-12 level specifically so that it may continue into undergraduate education in preparation for careers in engineering.

Background

In order to emphasize engineering information principles, the team needed to select a focus domain with projects to hold the attention of the students and excite them to:

- proactively participate and bring out their competitive spirit,
- design projects that are multi-disciplinary in nature but have the manageable level of complexity for student projects, and
- design intelligently with information on working principles of the underlying components, the design requirements, and availability of underlying components. Biologically-inspired robots meet all of these criteria and hence are expected to serve as a very good choice for the design project domain.

[1] http://gicl.cs.drexel.edu/wiki/CI-TEAMS:_Engineering_Informatics_for_Bio-Inspired_Robotics

Bio-inspired robots are a new frontier in autonomous systems. The CI-TEAM used this domain to broaden the appeal of computing and engineering. Some types of applications that are not suitable for traditional robotic systems, ie. certain types of terrain, are more suitable for bio-inspired robots. For example, search and rescue missions in complex urban environments require devices that can maneuver in collapsed buildings (e.g., cockroach), ductwork (e.g., ant), and other obstacles (e.g., lizard, centipede, etc). A snake-like robot may be used for planetary surface exploration, minimally invasive surgery, or inspection of piping and cabling.

There are a multitude of informatics challenges for these devices. Designing bio-inspired robots is a highly multi-disciplinary activity. Components include sensors, actuators, structural components, electronics, power source, and software. The actuated joints can create an extraordinary number of degrees of freedom (DOF). Joints must operate individually or be coordinated centrally; motion-planning algorithms must be rethought; and physics-based modeling and simulation tools need to scale to handle vastly greater complexity. For example, design and simulation of a snake inspired robot leads to numerous problems: these devices can include thousands of components, nearly all of which must interact for the snake to work. Additionally, one must capture engineering phenomena across all disciplines: mechanical, electrical, chemical, computer science, electronics, and environmental. The combination of scale and domain complexities demands extensive number crunching that requires highly optimized tools and efficient algorithms.

The CI-TEAM's goal was to create lasting integrations across the disciplines, both in the classes and in the students. A specialized set of course materials were designed for use in undergraduate and graduate classrooms, but this facilitates derivative products for use in outreach and for production of on-line materials, making the modules available to other students as well.

The Challenge of an Engineering Informatics Education

Many modern engineered products require extensive use of informatics principles. These principles need to be applied at two different levels. First, appropriate engineering models need to be created throughout the design process to support design, analysis, and simulation. When a system being designed spans multiple domains (e.g., software, mechanical structure, electronics), different types of individual engineering models and simulations need to be integrated together to support the design process. Second, many engineered products today include embedded software. This software needs to be synthesized from existing software components and managed as a part of the overall design process. Inefficiencies in engineering model construction can significantly lengthen the product development time and lead to suboptimal product performance.

One of the goals of the CI-TEAM Demonstration project was to conduct an exploratory educational experiment to investigate these hypotheses. The goal of the CI-TEAM Implementation project was to scale these efforts and transfer the knowledge and pedagogical materials to a national audience.

Project Theme

Biologically-inspired robots fascinate students from a wide variety of backgrounds and appear to be a good candidate to not only hold student attention but even attract students to science and discovery. Abundant sources of information are available on how to design these robots and they can be built as projects during the course of an academic semester or term. However these sources are mono-disciplinary and are not, on their own, a suitable basis for an informatics education. The challenge is that these robots are highly multi-disciplinary consisting of sensors, actuators, mechanical structures, electronics, and software. Components from these diverse disciplines need to be tightly integrated to create successful robots. The challenge to the student roboticist is to develop a working understanding of each of the required disciplines—an understanding adequate enough to appreciate (and debug) the complex issues that emerge at their interfaces.

Related Work

There are several other related efforts with which this project synergizes. For example, the NSF GK-12 Program is run at many universities, including Drexel, with an aim to inspire students towards science disciplines. Particularly, the Drexel GK-12 mission is to use engineering as a contextual vehicle for teaching math and science, with a goal of educating students about and motivating them to pursue engineering disciplines.

The following are the main categories that are relevant from an information modeling point of view:

- Design Information Flow Models: This area deals with the modeling of the information being generated as the requirements are mapped into detailed description of products. Representative work includes (S. Shooter, W. T. Keirouz, S. Szykman and S. Fenves 2000; S. Szykman, R.D. Sriram and W. C. Regli 2001; S.J. Fenves 2001).
- Requirements Modeling: Extensive research was performed on modeling of requirements, and is usually concerned with modeling the requirements that drive the design process. Representative work includes (S. Sunnersjo, I. Rask, and R. Amen 2003; M. W. Fu and W. F. Lu 2003; B. Becker and N. Wang 2003; L. Balmelli and A. Moore 2004).
- Function Modeling: Determination of functions and performing functional decomposition play an important role during the conceptual design. Representative work includes (C. F. Kirschman and G. M. Fadel 1998; R.B. Stone and K.L. Wood 2000; J. Hirtz, R. Stone, D. McAdams, S. Szykman, and K. Wood 2002; M. R. Bohm and R. B. Stone 2004).

- Conceptual Design Modeling: This area mainly deals with modeling of designs during the conceptual design stage. These types of representations mainly try to capture underlying components, their functions, and their behaviors in an integrated framework. Representative work includes (Umeda & et al 1996; Vargas-Hernandez & Shah 2004; X. Fischer, C. Merlo, J. Legardeur, L. Zimmer, and A. Anglada 2004; Xu, Gupta, & Yao 2004).

Cyber-Infrastructure for Engineering Informatics Education

The CI-TEAM program, the bio-inspired robots repository and its engineering education connections were created via a series of courses offered at participating universities, including Drexel University.

Bio-Inspired Robotics in Support of K-12 Educational Programs

The initial course at Drexel was executed in the Fall quarter of 2006. This course was executed as a seminar class, with 12 students spanning computer science, mechanical engineering, computer engineering, software engineering and electrical engineering taking the class. The goal of the class was to introduce students to "engineering informatics" while building comprehensive engineering models of biologically-inspired robotic systems. Each student had to develop his/her own design of a bio-inspired locomotion mechanism, build it with a robot construction kit (most chose to use Lego Mindstorms kits) and implement a complete 3D model with kinematics simulation and, if possible, physics. In doing so, students learned to:

- identify problems resulting from the interdisciplinary interactions in bio-inspired robots;
- perform system engineering to design, test and build bio-bots;
- apply informatics principles to bio-robot design and testing;
- use pedagogically appropriate hardware (i.e. Lego Mindstorms, Roombas, etc) and software tools for robot design/analysis.

Snake-Based Robot

Among the most successful projects is the Bio-Inspired Snake-Based Robot (Figure 1). The goal is to design a robot that behaves similar to a biological snake. The robot is a good platform for testing various control, path planning and object avoidance algorithms. The student constructed the physical device and built motor drive electronics.

ADAMS was used to model the snake-based robot, including Motor Segments, Board Segments, and Frames, such as the Board Segment seen in Figure 2. The model was tested using control algorithms written in MATLAB. As with the other projects, it is more feasible and more cost effective to test and experiment with a software model than a physical one.

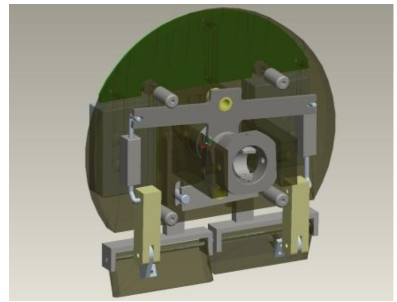

Figure 2: Bio-Inspired Snake-Based Robot Board Segment.

Roomba Design Lab

As part of the 2006-2007 Freshman Engineering curriculum at Drexel University, all 650 freshmen in the College of Engineering were required to take an interdisciplinary Design Laboratory. There were 6 design lab units and they varied across disciplines. A set of laboratory exercises were designed for the iRobot Roomba platform. There were two main objectives for the Roobma Module:

- To explore the interaction between software and information and the physical world by animating, via software, a robotic "creature." The robotic platform had aspects that pertain to all of the engineering disciplines in the Drexel program.
- To use cyber-infrastructure to obtain and share the information needed to execute the module. In the course of this laboratory assignment, no paper was distributed and all information regarding project deliverables and infrastructure was provided through the project Wiki. This also included a "Roomba Questions" email list, Roomba Questions message board, and web resources for programming and experimenting with Roombas.

The Roomba Lab was executed over a 4 week period with students working in teams of 3-to-5 students per team. The assignments were designed to be accessible to freshman undergraduate engineering students while still requiring students deal with the interaction between software, sensing the physical world and the robotic platform itself. The lab cumulated with a contest to determine which team was able to identify and implement the best method for dealing with each of three robot maze challenges. The use of mazes as challenges for robots can be traced to early work by Sutherland and the current work of NIST for design of urban search/rescue robotics courses. The challenges to be executed as part of the lab included:

- **Challenge #1: The Chutes.** Given a detailed and exact layout of the world, with single exit path and no possible wrong turns or shortcuts, implement a set of instructions (i.e. a path) for the Roomba so it can navigate the start to the finish. The Performance ob-

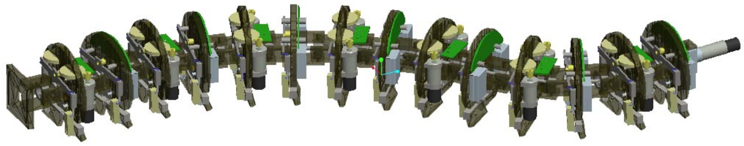

Figure 1: Bio-Inspired Snake-Based Robot.

jective is to minimize time or the resulting minimize distance to finish line.

- **Challenge #2: The Maze.** A few hours before the competition, a new maze is released. The Roomba is placed in this maze in a pre-determined starting location. Assuming there is only one path out (but there may be dead-ends and detours) students needed to implement an algorithm to find a path out of the maze, preferably by writing code for general maze solving (but given that students have access to the maze configuration, hard coding the path via dead reckoning is possible). The objective was to minimize time and minimize distance within the maze to the finish line.
- **Challenge #3: The Great Escape.** The Roomba is placed in an arbitrary maze in an arbitrary starting location. Due to the nature of the challenge, the only solution that should work is a general maze solving strategy. The objective was to minimize time and minimize distance within the maze to the finish line.

For the scenarios when the path and world state is explicit, solutions have to deal with error and inaccuracy. For example, the dead reckoning on the Roomba accumulates errors and students needed to compensate for these; if walls are bumped, the Roomba can further loose track of its location. Students had to account for error introduced by the movement of the vehicle, friction, uncertainty in orientation and in the initial set up. Further, students needed to use the Roomba sensors when developing strategies to handle dead-ends, non-rightangled walls, and short cuts that are hard to sense. Lastly, teams also had to address the trade-offs inherent in various behaviors. For example, a quicker algorithm may not be guaranteed to find the solution while a slower one might.

Enhancing K-12 Engineering Education through the Study of Aeronautics and Bio-Inspired Flight Robotics

One bio-inspired robot project intended for K-12 education is the flight-capable "bio inspried robots." Instructional goals include exposure to the principles of aerodynamics and aeronautics, evolution of and differences among powered flight, and the history of human flight as inspired by nature. This project was inspried by a morphing air-vehicle which cranes similar to an "organic flight vehicle" (Abdulrahim 2004).

For the course project, software simulation models are explored for impact in a K-12 environment - specifically, principles of aeronautics were modeled in a computer simulation environment and prepared as part of a multi-disciplinary elective on engineering for middle grade students as part of the NSF GK-12 Project. Because of this target audience, particular emphasis was placed on flight models of "robot airplanes" that are capable of navigating confined spaces, such as indoor hallways. Particular areas of interest are turning radius, size / payload constraints, and minimum controllable airspeed. For example, it would be ideal to discuss the flight characteristics of a small flying object that operates indoors and throughout hallways. In this way, course content is again presented as a series of challenges - that is, can students construct an aircraft that can satisfy various characteristics, such as staying in the air for certain periods of time or traveling certain distances. This open-ended description alone naturally lures students into a loggable trial-and-error process that involves varying components in the aircraft through inquiry and the scientific method. The airplanes were modeled using Athena Vortex Lattice (avl) (AVL), an MIT windtunnel simulation package for an arbitrary geometry. AVL allows for a text-based definition of the airfoil geometry.

Simulation For the simulation, we took a simple aircraft included in the AVL package (see Figure 3), and observed its flight characteristics in a 30 to 45 degree turn. Because a turning aircraft requires a horizontal lift component, some of the aircraft's lift is taken from the vertical component that keeps it in level flight. Therefore, it is necessary to apply up elevator on the aircraft to introduce additional vertical lift to compensate. However, the wings and elevator are so small on this aircraft that a considerable amount of elevator pressure is required to keep this aircraft level. It is not likely that the aircraft could sustain this otherwise normal maneuver.

As a proof of concept, the airframe was modified to include a larger elevator, a triangular body (this was used to make specification easier, as the body of the aircraft is irrelevant to its flight characteristics here), and larger wings. This aircraft was put in a slow turn to simulate turning a corner in a hallway, which is a possible real-world scenario of small, unmanned, indoor surveilance aircraft. This aircraft was able to make the turn using shallow banks of 10 degrees or less, making it less important to apply any significant up elevator

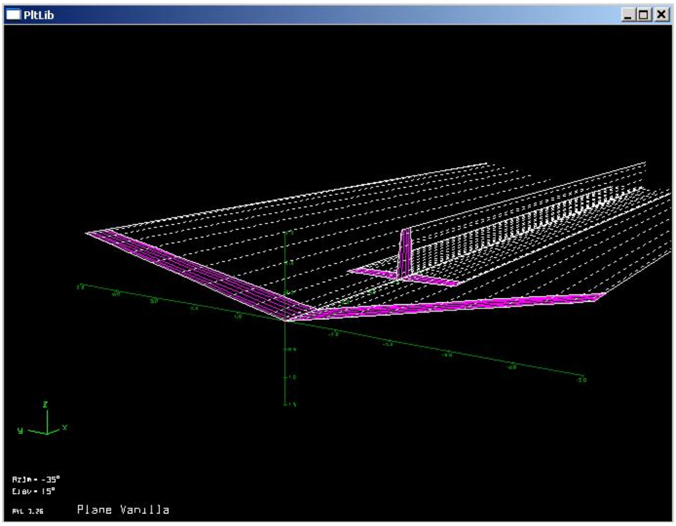

Figure 3: Vanilla airplane included in the AVL package for demonstration.

to maintain altitude. These results can be seen in this video still in Figure 4.

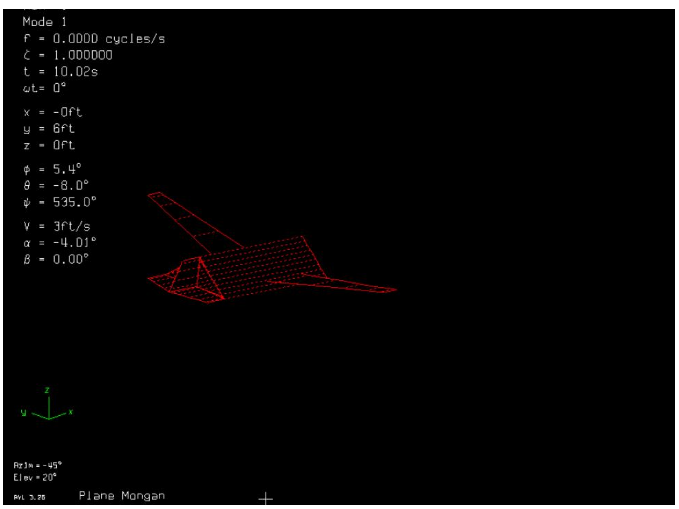

Figure 4: The modified airplane in a shallow turn at low airspeed, without loss of altitude.

Classroom Applications This simulation was intended for use in the classroom. Flight is itself bio-inspired, and each provides a grounded engineering education while also inspiring students from K-12 to undergraduate study. For these lessons, students first create or use a small aircraft from balsa wood or even paper, and try to make it fly. They should observe that subtle differences between each airplane causes noticable differences in flight characteristics, leading to a discussion of the parts of an airplane and their contributions to flight. Rather than continuing to throw their airplanes, students constructed a wind tunnel for streamlined testing and experimentation. Using index cards, they will also re-create the wing and tail control surfaces of the airplane and manipulate them to achieve pitch, yaw, and roll. They will also experiment with various sizes of control surfaces and their impacts on flight characteristics. Using the AVL package discussed in Section , students modify the sample airplane into a model that satisfies certain constraints (i.e. "able to corner about a three foot radius"). AVL produces data graphs for more advanced students, and video results for primary students to visualize their results, or make changes as needed.

Conclusion: Impacts on Education

The CI-TEAM program consists of a number of projects at the undergraduate and graduate level to create and maintain a repository of bio-inspired robotics for use in engineering informatics education. The robots and concepts explored were experimented with and used in course curricula at both the K-12 and the undergraduate level, including middle grades study in the School District of Philadelphia, and the freshman engineering courses at Drexel University.

The Role of Cyber-Infrastructure

A Wiki was chosen for a number of reasons as the central infrastructure for this project. First, it provides easy and universal access both for the end-user as well as for the contributor. Moreover, the Wiki removes the distinction between the consumer and the contributor, allowing the student to contribute to and build the repository, and more directly connect to the engineering disciplines and concepts. Because a Wiki is well understood by students throughout K-12 and college, it provides a more transparent and less intimidating medium through which students can become engaged with engineering disciplines. These modules are made available with the hope that students with negative impressions about engineering disciplines can access this information and become excited about the challenges and possibilities contained therein.

Acknowledgements

This work was supported by the National Science Foundation Office of Cyber-Infrastructure through grants OCI-0636235, OCI-0636273, SCI-0537125 and SCI-0537370.

The author is supported by NSF grant DGE-0538476.

Additional support was provided by the Library of Congress' National Digital Information Infrastructure and Preservation Program (NDIIPP) through the National Science Foundation (NSF) DIGARCH Program Grant IIS-0456001.

Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the

National Science Foundation or the other supporting government and corporate organizations.

References

Abdulrahim, M. 2004. Dynamic characteristics of morphing micro air vehicles. Master's thesis, University of Florida.

Athena vortex lattice. http://web.mit.edu/drela/Public/web/avl/.

B. Becker and N. Wang. 2003. ERMM: An Engineering Requirements Management Method. In *In Computers and Information in Engineering Conference*. Paper CIE-48238, Chicago, Illinois.

C. F. Kirschman and G. M. Fadel. 1998. Classifying functions for mechanical design. *ASME Journal of Mechanical Design* 120(3):475–482.

Cicirello, V., and Regli, W. C. 2002. An approach to a feature-based comparison of solid models of machined parts. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM)* 16(5):385–399.

Hayes, E. E.; McWherter, D.; Regli, W.; Sevy, J.; and Zaychik, V. 2000. Software architecture to facilitate automated message recording and context annotation. In *Network Intelligence: Internet-Based Manufacturing, Proceedings of the SPIE, Volume 4208*. Boston, MA: International Society for Optical Engineering.

J. Hirtz, R. Stone, D. McAdams, S. Szykman, and K. Wood. 2002. A functional basis for engineering design: reconciling and evolving previous efforts. *Research In Engineering Design* 13(2):65–82.

L. Balmelli and A. Moore. 2004. Requirements Modeling for System Engineering Using SYSML, The Systems Modeling Language. In *In Computers and Information in Engineering Conference*. Paper CIE-57751, Salt Lake City, UT.

2003. Long term ecological research, workshop report on environmental cyberinfrastructure needs for distributed sensor networks. http://intranet.lternet.edu/archives/documents/reports /sensor_report/cyberRforWeb.pdf.

M. R. Bohm and R. B. Stone. 2004. Representing Functionality to Support Reuse: Conceptual and Supporting Functions. In *In Computers and Information in Engineering Conference*. Paper CIE-57693, Salt Lake City, UT.

M. W. Fu and W. F. Lu. 2003. Modeling and Management of Design Requirements in Product Deveopment Life Cycle. In *In Computers and Information in Engineering Conference*. Paper CIE-48236, Chicago, Illinois.

McWherter, D.; Peabody, M.; Shokoufandeh, A.; and Regli, W. 2001. Solid model databases: Techniques and empirical results. *ASME/ACM Transactions, The Journal of Computer and Information Science in Engineering* 1(4):300–310.

NCAR. 2003. National center for atmospheric research, workshop on cyberinfrastructure needs for environmental research and education. http://www.ncar.ucar.edu/cyber/cyberreport.pdf.

Piasecki, M.; Amin, M.; Dyke, S.; Lin, M.; Neumann, U.; Rawlings, J.; Spencer, B.; Rao, V.; and Nelson, P. 2004. A report on the 3rd nsf-engineering directorate cyberinfrastructure workshop, research opportunities in cyberengineering and cyberinfrastructure development. Technical report, Drexel University, Philadelphia. http://thor.cae.drexel.edu/ workshop/Workshop_Report.pdf.

R.B. Stone and K.L. Wood. 2000. Development of a functional basis for design. *Journal of Mechanical Design* 122(4):359–370.

Regli, W. C., and Cicirello, V. 2000. Managing digital libraries for computer-aided design. *Computer Aided Design* 32(2):119–132. Special Issue on *CAD After 2000*. Mohsen Rezayat, Guest Editor.

S. Shooter, W. T. Keirouz, S. Szykman and S. Fenves. 2000. A model for information flow in design. In *In ASME Design Theory and Methodology Conference*. Baltimore, Maryland.

S. Sunnersjo, I. Rask, and R. Amen. 2003. Requirement-Driven Design Process with Integrated Knowledge Structures. In *In Computers and Information in Engineering Conference*. Paper CIE-48218, Chicago, Illinois, USA.

S. Szykman, R.D. Sriram and W. C. Regli. 2001. The role of knowledge in next-generation product development systems. *Journal of Computation and Information Science in Engineering* 1(1):3–11.

2003. Scalable information networks for the environment workshop. http://pbi.ecoinformatics.org/sine_workshop_report.html.

S.J. Fenves. 2001. A core product model for representing design information. Technical Report NISTIR6736, National Institute of Standards and Technology, Gaithersburg, MD, USA.

Umeda, Y., and et al. 1996. Supporting conceptual design based on the function-behavior-state modeler. *AIEDAM* 10(4):275–288.

Vargas-Hernandez, N., and Shah, J. 2004. 2nd-cad: a tool for conceptual systems design in electromechanical domain. *Journal of Computing and Information Science in Engineering* 4(3):28–36.

X. Fischer, C. Merlo, J. Legardeur, L. Zimmer, and A. Anglada. 2004. Knowledge Management and Support Environment in Early Phases of Design Process. In *In Computers and Information in Engineering Conference*. Paper CIE-57791, Salt Lake City, UT.

Xu, C.; Gupta, S.; and Yao, Z. 2004. A framework for conceptual design of multiple interaction state mechatronic systems. In *Tools and Methods of Competitive Engineering Conference*.