Combined Research and Curriculum Development in Information-Centric Systems Engineering

Mark A. Austin, John S. Baras, and Natalya I. Kositsyna
Institute for Systems Research, University of Maryland, College Park, MD 20742, USA.
E-mail: austin@isr.umd.edu, baras@isr.umd.edu and kosnat@isr.umd.edu.

Abstract. This paper describes a multidisciplinary program of research and curriculum development in Information-Centric Systems Engineering currently underway at the Institute for Systems Research, University of Maryland, College Park. This work is supported by a significant grant from the National Science Foundation and industrial funds from US industry. After motivating the need for an information-centric approach to systems engineering education, this paper describes the research methodology, pathway from research to curriculum development and teaching, and architecture for web-based project deliverables.

INTRODUCTION

Over the past fifty years we have witnessed unprecedented and rapid development in systems engineering. Most of these advances have been achieved following a strong interplay between the sciences and engineering, mostly on a reductionist basis. As a result, many recent discoveries have focussed on individual components and have emphasized narrowness and depth in engineering and engineering knowledge. These developments and evolution are most evident in current education curricula with the well known weaknesses of "fragmentation" and "over specialization." During that past decade several prominent reports from the National Science Foundation, National Research Council and National Academy of Engineering have emphasized these problems and called for and engineering curriculum revision that emphasizes a more integrative approach to engineering, more emphasis to connections among the disciplines, a more-wholeistic or systems view (Dowell, 1994; NAE, 1996; NRC, 1995; NSF, 1996). In other words, these prominent reports have called for more emphasis on teaching synthesis. Synthesis in engineering is the principal objective of systems engineering, and it is precisely the subject of teaching synthesis that forms the basis for combined research and curriculum development in systems engineering

One may post the question as to why this increased emphasis in teaching a "systems view" or synthesis is now appropropriate? In addition to the philosophical and pedagogical correctness with respect to the subject of engineering as a whole, over the past fifteen years there have been several important reasons and developments that have rendered such educational programs and methods critical. They are: (1) Rapid changes in technology; (2) Fast time-to-market critical; (3) Increasing pressure to lower costs; (4) Increasing higher performance requirements; (5) Increasing complexity of systems and products; (6) Increased presence of embeded information and automation systems; and (7) Failure due to lack of systems engineering. 70% of product and system failures are due to no or bad systems engineering effort, as our industry advisors and collaborators have frequently stated

Master of Science in Systems Engineering (MSSE). In 1987, the Institute for Systems Research (ISR) at the University of Maryland, College Park, created a broad-based academic program in Systems Engineering Education at the Masters Level. ISR's program in education compliments its mision of research towards the full integration of control, communications and computations for the rapid design and production of durable, versatile and affordable engineering systems.When the MSSE program was first created our educational focus was on the preparation of young people for a first career position. During the past 5 to 7 years, however, major shifts have occurred in the business and engineering landscape, and we now find ourselves educating large numbers of mid-career individuals (most of them working full time) wishing to balance their professional experience with academic training in systems engineering. In a first step towards ensuring our courses are relevant to both student sectors, our faculty have experimented with cross-listing courses across university departments, and delivering short course summaries to industry. This project will elevate this effort to an entirely new level.

KEY CHALLENGES

These developments have bought about a new environment in which engineering is practiced and systems engineering is learned. The consequences of these facts are enormous and present major challenges to the practicing engineer. In more detail, the key challenges are:

Synthesis from Modular Components. The prevalence of synthesis from modular components is no longer true just for aerospace, defense and large government contracts. Instead it is required in all commercial designs and operations. This so-called "systems integration" has become key and perhaps the most profitable engineering practice.

Support for Team Development. A key challenge is avoiding communication and interpretations problems that can occur when teams of experts from multiple disciplines/domains work together to solve a complex industry problem.

Growing Importance of Information-Driven Systems. In the past, systems have been seen from an operations point of view, where information and communications have been regarded as the supply of services necessary for the system to operate in predefined ways. Nowadays, there is a rapidly evolving trend towards the team development of large-scale information-dominated systems, which exploit commercial off-the-shelf (COTs) products and communication technologies, and are derived in response to various types of information from a wide array of sources.

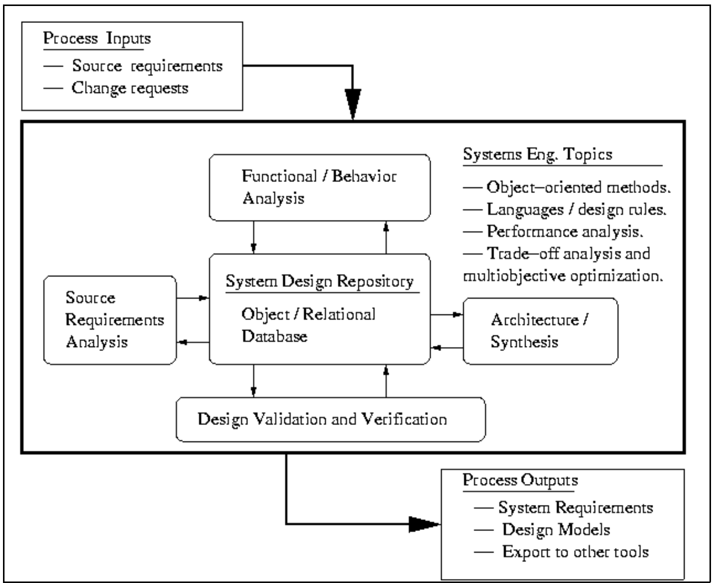

Figure 1. Information-Centric Systems Engineering

Large Volumes of Heterogeneous Data. The characteristics of scientific, technical and business data is further amplifying these challenges. Current and future data sets are in large volumes (not all relevant), numerically intensive (often requiring parallel algorithms for processing), multidimensional and heterogeneous, and present multiple views to the various users (e.g., engineering team members; marketing people; sales people; management and customers).

Our Vision. The foundation of our vision for systems engineering research and education lies in the use of information representations, models and advanced techniques for manipulating information. We have coined the phrase "Information-Centric Systems Engineering" for this effort, which will contain at its heart the processes and activites shown in Figure 1.

ELEMENTS OF INFORMATION-CENTRIC SYSTEMS ENGINERING

Based on more than ten years of research at the ISR and our MS in Systems Engineering, we have distilled the following items as key principles and methodology in our strategy and approach to Systems Engineering practice:

1. Promote information-based design, operation and management of systems, products and organizations.

2. Do everything using computers and information abstractions, from conception, to design, to parts selection, to manufacturing, to operations and retirement.

3. Hardware and software implementations and specific technology selection are delayed as long as possible, but are performed once and must work flawlessly.

4. Abstract multiple disciplines to properly annotated information abstractions. This is the only way to allow communication among disciplines and multiple contextual views. This approach also facilitates much better management of the overall process.

5. Develop sophisticated algorithmic, mathematical and quantitative methods implemented in modern software environments.

6. Emphasize function-architecture co-design. This involves working simultaneously on top-down methodology and bottom-up (specific applications) research and advances.

Our educational goals are to promote these key principles together with emerging methodologies for the design of complex systems – the latter include an emphasis on orthogonalization of design concerns; strategies for re-use at all levels of abstraction, and use of formal methods for the representation of system specifications.

PROJECT COMPONENTS

Key Technical Areas. The key technical areas that we have identified as being a critical need for more systematic training and education are as follows:

1. Object modelling of systems using the Unified Modeling Language (UML) and

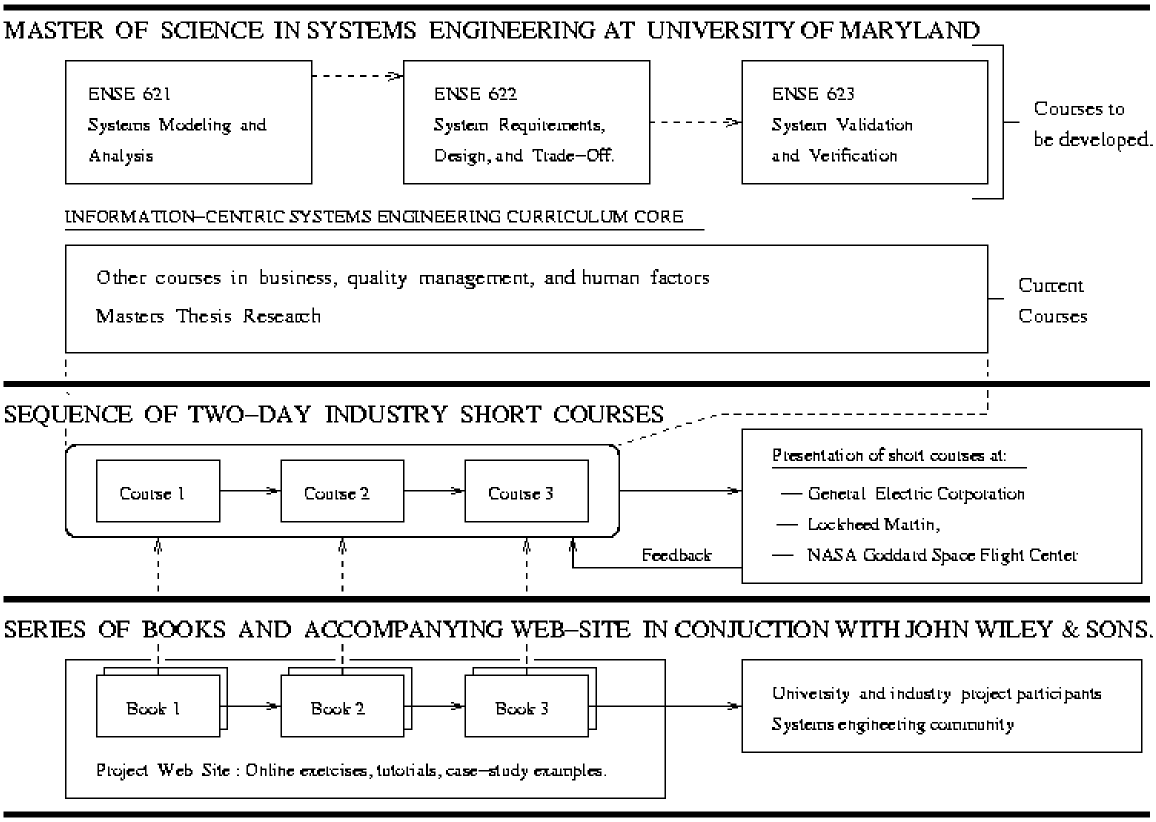

Figure 2. University, Industry and Publishing Components of Systems Engineering Education at ISR

2. automation of model-based system behaviour simulation (Booch et al., 1999).
3. Systems requirements and specification allocation in heterogeneous (from the physical layer perspectives) hierarchies.
4. Trade-off analysis when (mixed) Boolean and numeric values are present.
5. Validation and verification by quantitative treatement of tolerances and convex analysis.
6. Object-relational databases and multiple views (engineering and others) of system data.

To develop the needed methodologies we have selected four research projects that will be pursued in parallel and in tight coupling with the development of additional "knowledge modules" for our core Systems Engineering courses.

Linking Research to Key Technical Areas. We are working closely with our industry collaborators to develop and teach the new knowledge modules to targeted engineering audiences (small groups) within the participating companies (i.e., General Electric; Lockheed Martin and NASA Goddard Space Flight Center, as of November 2001). In an on-going iterative process, these "draft" educational materials will be evaluated by practicing engineers and provide feedback to further improving the material.

SYSTEMS CORE CURRICULAR

We are acutely aware that as a subject areas, systems engineering research and education depends critically on industry cooperation and feedback. In partnership with our industry advisors (i.e., General Electric, Lockheed Martin and NASA Goddard Space Flight Center) and John Wiley and Sons, our plans are to develop, critically evaluate and widely disseminate information-centric systems engineering curriculum. The relationship among these elements of work is shown in Figure 2. The five-part curriculum will be as follows:

ENSE 621: System Modeling Building and Analysis. Using data, information and knowledge relavant to an organization's measures of effectiveness, students will learn how to synthesize, articulate and refine the goals of an complex engineering system through the use of use cases, scenarios (i.e., fragments of typical system usage), generation of requirements from scenarios, which in turn leads into high-level simplified models of system behavior and system structure. Students learn how to

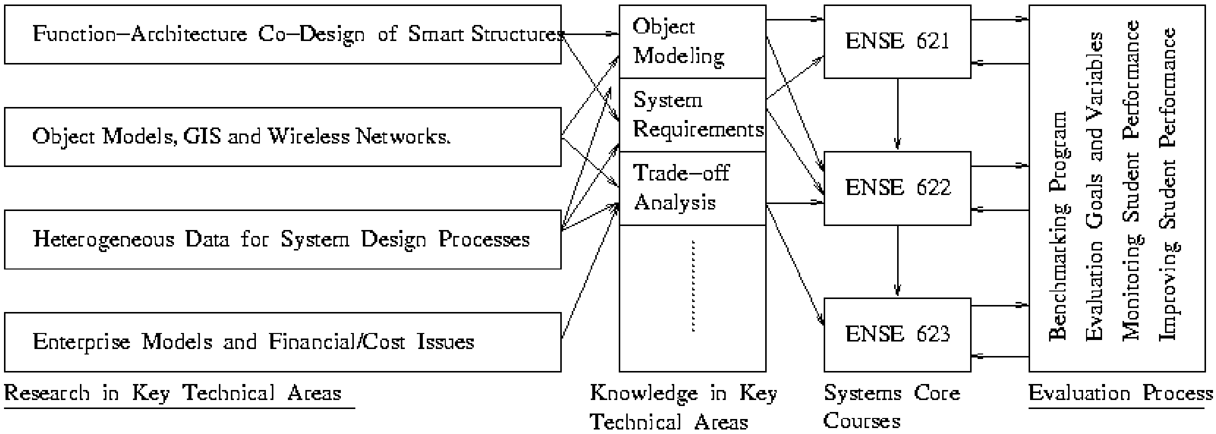

Figure 3. Linking Research in Key Technical Areas and Methodologies to Curriculum Development, Delivery and Evaluation

use object-oriented diagramming procedures (e.g., UML), decompositions, hierarchies and abstractions to describe the behavior, structure and information contained within their designs. A distinguishing feature of our curricular is the central role of object-relational database technology in systems engineering. The final topic in this course will be elements of object and relational databases, their interrelationships, and their use as depositories for systems, component models and data.

ENSE 622: System Requirements, Design and Trade-Off Analysis. Students will learn about requiremnts engineering, requirements traceability, co-design of system function and architecture, and methods for decision analysis and multi-objective trade-off analysis. Building upon the previous course and system models developed there, we will develop methods for capturing system- and component-level requirements within the object modeling approach and within the object-relational data model (Graham, 1998). Design specification languages, system behavior models, design grammars, will all be linked in such a way as to make cross-disciplinary teams capable of working within their specialty language, and at the same time preserve consistent traceability through changes made by any member of the team or members of the team. In the latter half of this course, students will learn how to model quantitatively requirements and specifications as either constraints or performance metrics. This will serve as an introduction to quantitative procedures for decision analysis, system optimization and multi-objective trade-off analysis.

ENSE 623: System Validation and Verification. This course will explore the benefits and conditions necessary for making validation and verification an integral part of system design, beginning in the earliest phases where corrections are easiest and cheapest to make. For strictly boolean systems automated processes for design verification and validation have been developed and are well known under the term "formal verification methods" (Manna, 1995; CAV, 1998). These utilize algorithmic tools that are only applicable to computer systems. Essentially, these methods are verification and validation methods for computer programs. Based on the information abstractions used in our systems engineering models it is possible to extend these methods so that they can be used for verification and validation of system behavior.

Graduate Systems Certificate Course. Ph.D. level graduate students will take the ENSE 621, 622, 623 sequence, plus three courses in their discipline containing systems content.

Industry Short Courses. A series of three, multi-day, web-based short course summaries of ENSE 621, 622 and 623.

Throughout these courses, out strategy will be to enable multi-disciplinary development and communications through appropriate information abstractions and representations. The pathway of development for curriculum development, evaluation in industry and university environments, and dissemination is shown in Figure XX.

RESEARCH PROJECTS

Research Projects. Development and delivery of these courses will be enhanced with a series of research projects designed to produce technical knowledge directly applicable to systems engineering education. Due to space restrictions, we will provide brief descriptions for only two of the research projects:

1. **Object Modeling, GIS and Wireless Networks.** In this research project we are investigating the construction of object models suitable for the systems integration of geographic information systems (GIS) and wireless/sensor networks.
2. **Function-Architecture Co-Design of Smart Structures.** In this project we are investigating the extent to which the Unified Modeling Language can be applied to the high-level description of smart structures

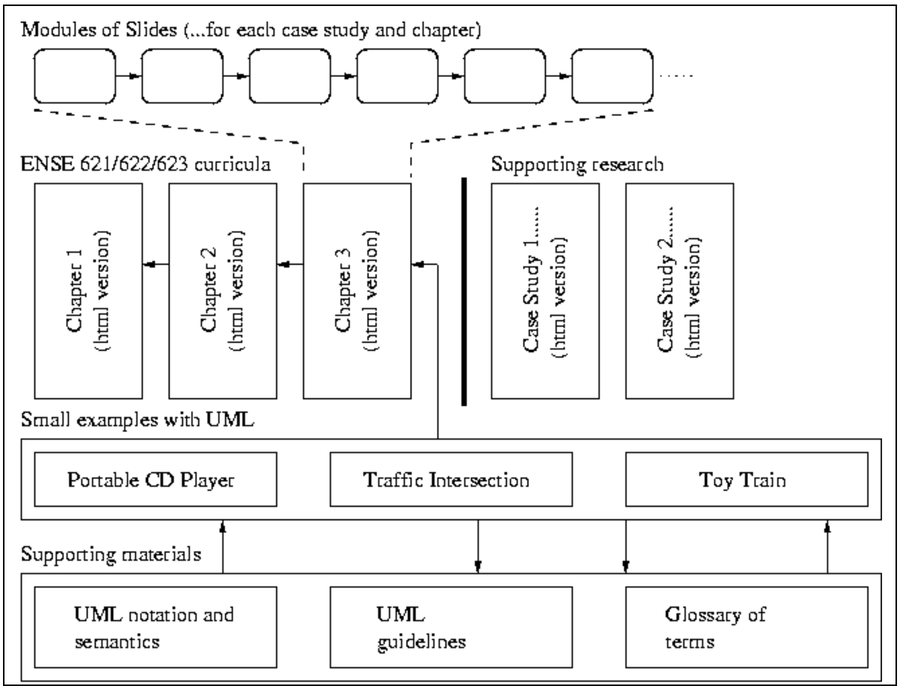

Figure 4 : Proposed Architecture for NSF CRCD Materials.

composed of structural, mechanical, and electrical elements.

The results from each research project will be weaved into the ENSE 621-622-623 course sequence, as shown in Figure 3.

ARCHITECURE FOR NSF CRCD MATERIALS

Web-based Architecture for Curricular and Project Deliverables. Figure 4 shows the proposed architecture for the NSF CRCD curricular materials. We expect that a large quantity of web-based curricular material will be developed during the three-year duration of this project, and as a starting point, we are developing a framework for the curricular format, and a layered approach to teaching object modelling of systems with UML.

Our goal is to publish books – see the lower sections of Figure 2 – that follow in the spirit of SUN Microsystems's series of texts on the Java Programming Language. First we will develop a web site containing comprehensive quantity of systems engineering material. As shown in Figure 4, this will include low-level support materials (e.g., a summary of UML notation and semantics; guidelines on how to use UML), small examples showing how UML and object modelling principles can be applied to the lifecycle development of everyday products and systems, web-based write-ups of the research results and, finally, curricula from the new versions of the ENSE 621-622-623 sequence.

We already know from our preliminary developments that some of the modules may be long (even very long). For example, graduate students in the MSSE program are currently working on the small case study examples. Developing a comprehensive case study for a CD player/walkman that includes use cases, goals and scenarios, initial and expanded requirements, and logical and physical designs can easily result in 100 pages of material. Hence, it is important to complement each case study with a web-based slide show that introduces the reader to the material, and provides pointers to more detailed explanations. Initial versions of our small-scale case study problems are being prepared in a standard html format – text, images, links, and so forth – that is rather linear. Once the content for each case study is in place, our plans are to reformat each study so that readers are introduced to the material in multiple layers of increasing depth and coverage. Case studies will also explore ranking and selection among design alternatives and this presents an ideal opportunity for small Java applet simulation programs to be weaved into the case study material.

Large-scale case study problems and research will be developed jointly with industry , and reported in the same format.

WEB-BASED SYSTEMS ENGINEERING

A challenge that we will address in this project is finding a practical way of using web technology to enhance: (a) classroom instruction, and (b) self-guided "post-training" instruction. This direction of work is motivated by training class experiences with our industrial partners, typically of two-to-five days in duration and containing hundreds of slides. Mechanisms for post-training follow-up are very poor. Without the guidance of the instructor on the relevant content in each slide, readers jump around and gloss over material rather than learn content. The result is low utility, in terms of what is learned and the rate at which learning occurs.

New Tools for Navigation of Web-based Training Content. It is evident that simply creating web-page equivalents of today's standard training materials will do little to mitigate the shortcomings of current models of systems engineering training. Our industry partners recognize this problem, and for the past few years have been supporting development of an expressive notation for bridging the gap between high-level learning objectives and lower-level systems engineering processes and task- and tool-oriented procedures.

To improve the efficiency of learning, and reduce the volume of training materials that a student will need to look at, lesson plan fragments will be indexed with specific learning objectives (e.g., a particular slide may contribute to learning objectives A and B, as shown in Figure 5). We will employ the "use case" construct to guide students with specific learning objectives through the content of individual diagrams and collections of diagrams. A use-case pathway is a wiggly line that enables a visualize scenarios threading through a system, without the scenarios being specified in great detail (Jacobson, 1992). From an educational perspective, the "use case" begins with a description of the learning objective, and is followed by an annotated pathway of tasks and activities that enable completion of the learning objective. Popup windows (i.e., textual and image hints) explain features along the pathway. Of course, more than one pathway can cross over a subsystem (indicating a need for communication and coordination of information), and how a single diagram can contain numerous pathways.

For the past two years we have been building a series of prototype environments to support this level of functionality. Our first prototype (Kositsyna, 2000) demonstrated that XML technology could be used to position and connect elements on a diagram, and implement a primitive form of "use case" pathways. The XML specification is then compiled into Java-enabled diagrams that are downloaded over the web as applets. While the XML-to-Java pathway is technically possible, it is now clear that the "low-level" generation of training materials by using a regular text editor will be far too complicated for most developers. Hence, in 2001-2002 we are developing a click-and-drop editor for diagram notation with pathways, hints, html links, and so forth (Kositsyna and Austin, 2001). Our project goal is to provide educators with an easy-to-use tool for creating and presenting highly interactive, colorful, and text-aded tutorial information on the web. Extensive knowledge of Java and the XML-to-Java compiler will not be necessary. Figure 6 is a screen-dump of Version 0.01 of our editor, and shows how a use case diagram can be created by simply clicking on the diagram element and dropping it on the canvas. The dialog box at the bottom of the figure prompts the content developer for a textual message that will appear in a popup message connected to the use case actor. Interactive Java-enabled diagrams are created by simply clicking on "Export to Java." It couldn't be easier! Our near-term development objective is support all of the UML diagram elements. Together, these features will allow content developers to create and annotate UML diagrams with pathways and popup hints, and link diagram elements to other entities on the web. With this capability in place, we foresee the development of drag-and-drop systems tools that can setup networks and hierarchies of UML diagrams for system specification, behaviour and structure, and provide connectivity to domain-specific content (e.g., engineering simulations; tools for system optimization and trade-off analysis). Looking further ahead, future versions of the training enviroment might also permit students to add their own annotations in much the same way that they scribble on today's paper training materials.

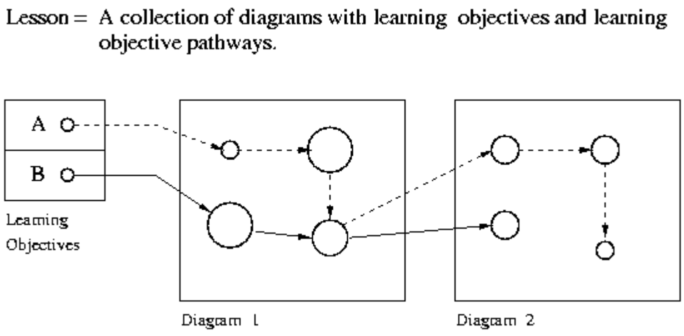

Figure 5. Learning Objectives and Pathways weaving through Collections of Diagrams

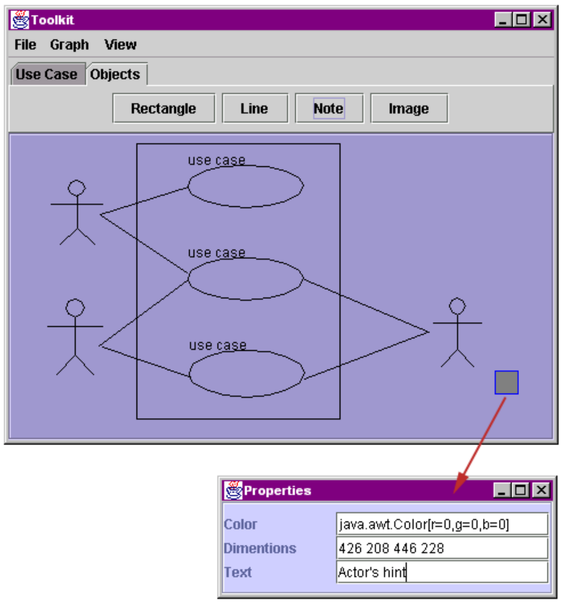

Figure 6. Screendump of our click-and-drop editor, Version 0.01.

CONCLUSIONS

Where are we? At the time of writing (November, 2001) we are at the end of year one in a three-year NSF funded project for Combined Research and Curriculum Development in Information-Centric Systems Engineering. . Our next task is to begin creating tutorial content, identifying key learning objectives and mapping learning objectives to training materials.

During the next two years, we will advance our knowledge and understanding of systems engineering through focused research projects, prepare mountains of web-based curricular material on systems engineering, develop numerous case studies of medium and large size, and attempt to integrate everything together using the web-based tools described in the previous section. Advances in the tools and curricular material need to occur in parallel – since these ideas are new, and at this time are still untested, we expect that in the next few years many lesson and tool prototypes will be developed. We expect that deficiencies in the XML-based diagram markup language will be identified, thereby requiring extensions to the underlying diagram syntax. Our industry sponsors are acutely aware of this predicament and promise and are keen to jointly work on the dual track development.

Our long–term plan is to create an architecture for web-based corporate training that supports a library of general-purpose training modules linked to industry-specific case studies, engineering simulations, and procedures of online evaluations. See Figure 7. We envision that students will state their learning objectives and online training exercises will be dynamically configured to meet the students specific learning objectives. We believe that implementation of this functionality will require a combination of database, Java servlet (Hall, 2000) and XML technology.

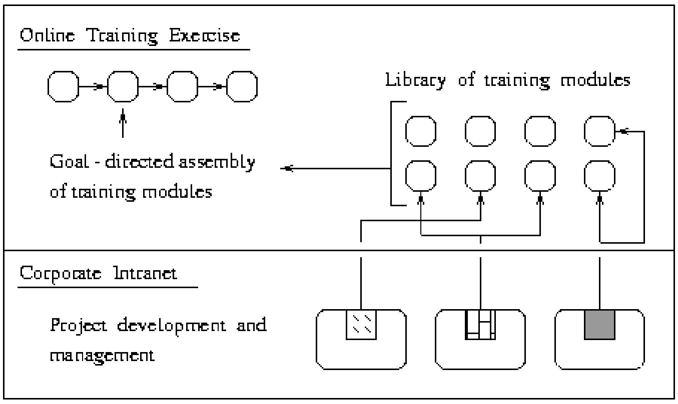

Figure 7. Architecture of Web-based Corporate Training Environment with Goal-directed Assembly of Modules

REFERENCES

Booch G., Rumbaugh J., Jacobson I., The Unified Modeling Language User Guide, Addison-Wesley, 1999.

CAV, Proceedings of the Tenth International Conference on Computer-Aided Verification, Springer-Verlag, 1998.

Dowell E., Baum E., McTague I., Engineering Education for a Changing World, A Joint Project and Report for the Engineering Deans Council, the Corporate Roundtable and American Society for Engineering Education, 1994.

Graham I, Requirements Engineering and Rapid Development, An Object-Oriented Approach, Addison-Wesley, 1998.

Hall M., Core Servlets and Java Server Pages, Prentice-Hall, 2000.

Jacobson R.E., Christerson M., Jonsonn P., Gunnar O., Object-Oriented Software Engineering – A Use-Case Driven Approach, Addison-Wesley, 1992.

Kositsyna N.I. and Austin M.A., Annotated Diagraming (JaDia) Web Site: see http:www.isr.umd.edu/~kosnat/tutorial/, December, 2000.

Kositsyna N.I. and Austin M.A., Click-and-Drop Java-enabled Diagrams Technology Web Site : http://www.isr.umd.edu/~kosnat/ToolKit/intro.html, November, 2001.

National Academy of Engineering (NAE), Academic Engineering Research in a Changing World, 1996.

National Research Council (NRC), Graduate Education in Engineering and Science, Committee on Science, Engineering and Public Policy, study, 1995.

National Science Foundation (NSF), Restructuring Engineering Education: A Systems Approach to Integrated Curricula, 1996.

SUN Microsystems, Java Programming Language, California, 1995.

BIOGRAPHY

Mark Austin Associate Professor at the University of Maryland College Park, with joint appointments in the Department of Civil and Environemntal Engineering and the ISR.

John Baras. Professor at the University of Maryland with joint appointments in the Department of Electrical and Computer Engineering and ISR

Natasha Kositsyna. Faculty Research Assistant, ISR.