Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

FOCUSED ANALYSIS AND TRAINING ENVIRONMENTS

F. Bradley Armstrong

ABB Power T&D Company, Inc.
1021 Main Campus Drive
Raleigh, NC 27606 USA

Barbara Werner Mazziotti
Michelle Benjamin

Textile/Clothing Technology Corp.
211 Gregson Dr.
Cary, NC 27511

Ken Powell

Simulation Solutions
6929-20 Springcreek Cove
Raleigh, NC 27613 USA

ABSTRACT

This paper introduces the concept of a Focused Analysis and Training Environment (FATE) as a logical evolutionary step for analysis techniques such as simulation. The intent of a FATE is to provide complete analysis capabilities to untrained users responsibly. A FATE consists of a sophisticated, graphical user interface that completely encapsulates an analysis tool such as simulation. The FATE concept exploits the fast construction times possible when using the new generation of software development tools by focusing on a very specific set of users and a subset of their problems. Narrow focus allows the FATE to control all aspects of use including data input, analysis, and reporting. The example included in this paper presents a prototype for the simulation-based subset of FATEs called Focused Application Simulator and Trainers (FAST).

1 INTRODUCTION

As simulation is pushed into the hands of end users rather than trained simulation users, the sole success criterion will be whether the tool will deliver answers at an "affordable cost" (i.e. time and money). Elegant models and entertaining animations, by themselves, will contribute little to establishing simulation as a mainstream analysis tool. Further, new users will have neither the time nor the interest in learning simulation skills. The availability of domain specific simulators (e.g. manufacturing, communications, and service) that offer animation and data-driven model construction have accelerated the spread of simulation, but users need more and are demanding it. A candidate list of product characteristics that will satisfy these new users (as well as avoid the current user "Shelfware" syndrome) might be the following:

1. Produces valid answers automatically
2. Does not require the user to have special skills
3. Is impossible to misuse
4. Requires no training
5. Requires minimal effort by the user
6. Delivers answers quickly
7. Is affordable

While vendors have made significant progress towards these lofty goals, no current products can deliver everything on this wish list. Beyond stating that a need exists, the purpose of this paper is not to debate specific issues, but rather to present a new perspective and a way of satisfying this wish list through the use of FATEs.

The FATE paradigm recognizes that the model builder and model user may be different and explicitly allows separate interactions. As shown in Figure 1, the FATE concept goes beyond simply providing a turn-key analysis tool by breaking new ground in training as well. Serendipitously, analysis and training have a symmetry as defined by a FATE. The key difference between the Analyst/Instructor and the User/Student is data access. As part of the definition process, the Analyst/Instructor can specify a small subset of the data to be made available to the User/Student. We have applied the FATE concept for training and believe simulation-based training is a major growth area [Mazziotti, Armstrong, and Powell 1993]. However, the specific features of our FATE training work will require an entire paper to adequately describe; therefore, discussion of the training aspects of a FATE in this paper will be limited to specific details of the section 3 FATE example.

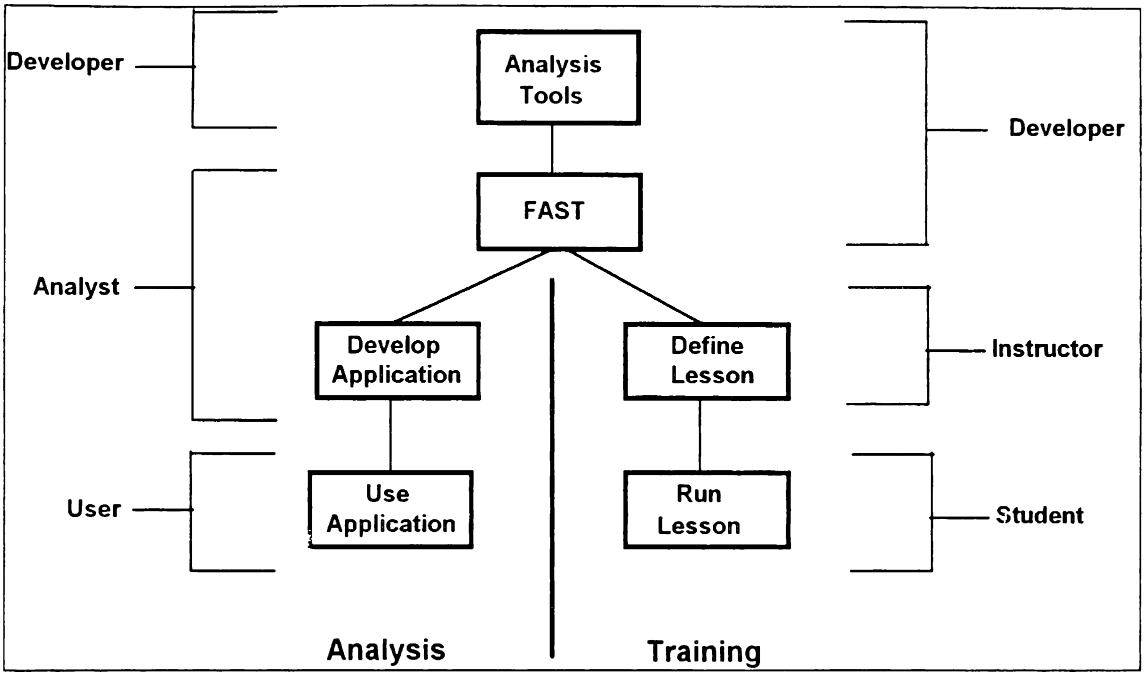

Figure 2: FATE Usage Structure

In the next section we describe and justify a FATE in terms of the presented wish list. In section three we describe a candidate FATE architecture, and in section four we discuss future work. In the final section we summarize and conclude.

2 FATE JUSTIFICATION

We present the following FATE justification from a simulation analysis perspective (bias) which means it actually justifies the FATE subset of Focused Application Simulator & Trainers (FATE/FAST). We believe this approach is appropriate for the intended audience. However, FATEs are not just for simulation analysis. Justifications for other primary analysis tools can be made and would have similar benefits.

2.1 Always produces valid answers automatically

The largest hurdle to responsibly providing simulation as a quantitative analysis tool to untrained users is the lack of statistical analysis design, execution, and interpretation assistance. Even for engineers who have had formal statistical training, mistakes are often made or inefficient designs used. Controlling the constructed model and the analysis performed ensures valid results. A focus‿d application allows the developer to identify the appropriate types of analyses and incorporate them into the interface. The number of supportable analyses is one defining factor for the amount of FATE focus required.

Data inaccuracies remain a problem (Garbage-In, Garbage-Out), but range checks can screen obvious problems. Additionally, a user may opt to use a "raw" data processor incorporated into the FATE application instead of specifying stochastic data directly. The processor takes "raw" system data and reduces it to an appropriate distribution and parameters. For chronological data, the processor also checks statistical assumptions such as independence and constant variance as well as judges the adequacy of available data from a statistical perspective. In essence, a FATE/FAST provides built-in statistical safeguards. Automatic functions present data to the user as suggestions that the analyst/instructor can override. Automatic functions should never be invisible to the analyst/instructor.

2.2 Does not require the user to have specialized skills

Modeling and statistics are the most common skills required of simulation product users. To meet this second requirement, then, the product must separate the modeling and statistics from the user which implies a data-driven approach. Unfortunately, without a modeling capability (through logic such as if-then-else, and variable/attribute constructs), flexibility suffers greatly. Application flexibility is required to represent variations between different systems or between different scenarios of the same system. However, for a sufficiently focused application, flexibility requirements will be minimal and can be incorporated into simple, data-driven choices.

By focusing the application, the number of menu levels and general complexity of the interface can be kept low. Many simulators, to support flexibility, use many menu levels and/or pop-up menus which can be confusing for beginners or infrequent users. A FATE, therefore, will not necessarily replace simulators, but may be used as a front-end to act as an additional filter between the user and the technology. The amount of flexibility and menu complexity act as constraints on the amount of FATE focus required.

2.3 Is "impossible" to misuse

The opportunities for misusing simulation when performing a project extend well beyond actual model building. Consider the following partial list of possibilities:

1. Simulation is not the right tool
2. Wrong objectives are selected
3. Inadequate/incorrect data used
4. Incorrect model built
5. Incorrect or no experimental design
6. Incorrect interpretation of results
7. Poor presentation of results

Avoiding these problems requires a knowledgeable decision maker. These decisions must be made when constructing the FATE which means the FATE must be built by a suitably trained individual. However, once built, a FATE analyst/user or instructor/student does not have to address these decisions other than to understand what they are and that they are consistent with their problem. A FATE clearly defines where and when untrained users can and should become involved in a modeling and analysis process. Also, the structure provided by a FATE will reduce user uncertainty and help focus activities on productive tasks.

2.4 Requires no training

The user will know her system and know what question she wants answered. Any other required knowledge can not be assumed known and must be given to the user via training The only way to guarantee a no training requirement is to provide an interface to the user which is immediately familiar and functionally obvious. Terminology is critical and should be user/plant specific. Industry specific terminology is often not acceptable because terms such as "part", "station", "process plan", etc. may be unfamiliar or familiar to the user in a different context than the model uses it. Also, menu levels are kept to a minimum and all data is accessible within three selections. Note that this simplified interface format requirement is a real constraint on the amount of FATE focus required. Lastly, extensive on-line help should be available for every decision and option a user must make to reinforce and reassure users. On the output side, analysis alternatives and outputs must be presented such that the user immediately understands the information being conveyed.

The development tools for a FATE interface emphasize adaptability in terms of terminology, color schemes, and layout. Fixed interfaces, regardless of how well they are designed, represent a barrier to user understanding and acceptance. The power of GUI development tools means users no longer have to settle for a generic interface.

2.5 Requires minimal effort by user

Users expend effort in three areas:

a. Data Input
b. Problem resolution
c. Results interpretation

One inescapable fact is that users will have to input some amount of data. However, a FATE reduces input time requirements in several ways. First, specific terminology means the user never has to spend time trying to determine what he is being asked to provide. Second, single keystroke accessible, context-specific help will immediately assist the user if a question does arise. Third, default values are included as much as possible, both for individual data fields and global specifications. Fourth, optional data is hidden until the need for it has been established. Fifth, menus are constructed to lead the user through the process and deny access to inappropriate areas. Forcing the user to follow a specific path constricts personal preference, but builds user confidence by minimizing uncertainty. Sixth, connections to computer-based data sources for automatic inputs are available for repeatedly populated FATEs.

All verification is done by the FATE. The user does not spend time confirming the consistency of the model and the accuracy of the input data. All data fields have type and range checks to ensure suitable input. A data entry problem is flagged immediately to the user. An overall model evaluation checks for consistency and flags problems to the user along with providing specific problem resolution information. Under no circumstances will the user be able to construct a model that will not run. In a focused environment, allowable/acceptable data is known much more precisely than would be possible in a more generalized product.

Outputs are tailored to the targeted users. The FATE report generator development tools focus on flexibility.

2.6 Delivers answers quickly

A FATE will greatly reduce turnaround time because scope, level-of-detail, modeling assumptions, model construction, analysis design, etc. are predetermined. Data collection will remain, but the FATE will automatically dictate exactly what data needs to be collected, and therefore provide concrete tasks that minimize user uncertainty.

Beyond the predefined project aspects built into a FATE, the number of runs required will also affect turnaround time. If the analyses (which are under the control of the FATE developer) are properly designed, a minimal number of runs will be required. If the application considers a range of systems, sequential designs will allow the analyses to be further tailored to a specific system.

2.7 Is Affordable

Affordability tends to be a function of many factors such as which industry and what need. However, by definition, if the poorest is satisfied all others will be satisfied as well. In this case, software running on minimally configured, standard equipment that does not need additional support software would be an idealized baseline. Further, the cost of mainstream PC/Workstation software packages define a baseline for the cost of the application software itself.

The expectation is that FATEs will become a mainstream analysis tool with a much broader customer population to spread development costs. The key from a development perspective is how quickly a FATE can be built. Methods for extracting user needs and preferences will have to be researched, tested, and formalized.

3 FATE DESCRIPTION

There are many reasons why a FATE is an attractive simulation application paradigm. The justification contained in the previous section focuses on the input and analysis benefits. However, there are additional benefits because a FATE seamlessly allows other analysis tools and training to be integrated. The best way to understand the power of the FATE paradigm is by looking at an example. The remainder of this section overviews a FATE Kanban Analyzer which will be used as a training aid for a JIT class at North Carolina State University [Armstrong, 1994]. As shown in Figure 2, the FATE consists of five major components. The remainder of this section describes each of these components. The interface was developed using the professional version of Visual Basic (version 3.0), the simulation and animation were developed using SIMAN/CINEMA (version 5.0), and the statistical support routines were developed using WATCOM C.

3.1 Files

This component interfaces the FATE with the operating system and shields the user. With two exceptions, all menu options are self-explanatory. The New option uses a predefined database that contains default data. In contrast, the Create option creates a new and empty database. The Create option is not shown/available to the students.

3.2 Data

The most progress has been made in this component. In the last few years, an explosion of GUI development products has provided developers with extremely powerful and easy-to-use tools to build sophisticated graphical user interfaces to manipulate data.

In general, the Data component controls the data input required to define the system to be studied. Beyond the control of individual inputs, data interrelationships can be cross-checked to eliminate the possibility of run time errors.

The System menu selection contains general system information and allows for general defaults to be specified. For example, the Kanban Analyzer allows the user to specify all or none of the Stations use a Transporter or Conveyor for material movement. If a system level default is selected, the Station input form will not ask for the defaulted information.

3.3 Analysis

This component is very flexible and rich in potential. In general, it evaluates, deterministically analyzes, and uses the information defined in the Data component. It is divided into two distinct parts: one for the Analyst and the other for the User.

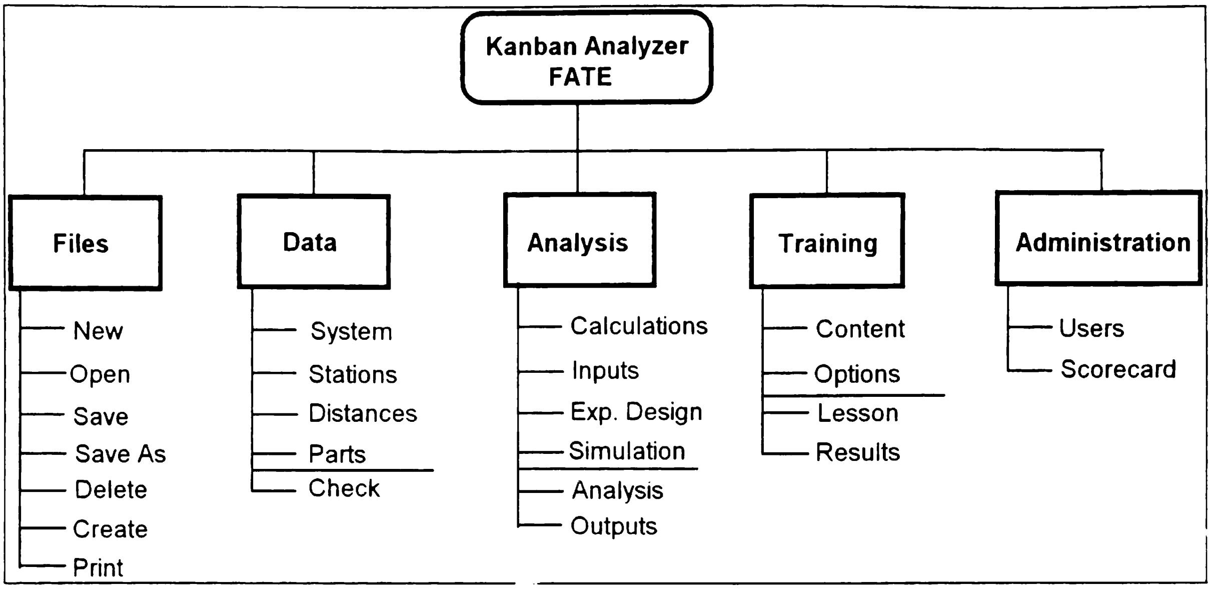

Figure 2: FATE Kanban Analyzer Organization

For example, as part of the data definition process for part operations, the user is given the option of either entering a value for the number of kanbans or requesting automatic calculation. The **Calculations** menu option in the Analyzer component triggers these calculations and presents the user with the results. These calculations represent sophisticated defaults that are based on other input data. Because this FATE was developed for class use, the user can ask to see individual calculations based on a "chalkboard" format. The user also has the option of overriding the calculation as long as the specified value is theoretically capable of attaining the stated goal. Providing automatic support while allowing user override is an important feature because it supports a wide range of user skills.

The **Inputs** menu choice is where the Analyst identifies the data the User will specify when performing an analysis. The inputs represent runtime data changes and will override data values specified in the Data component. The student is presented with an inputs form at run time but (in this FATE) can not interact with the model once the run has begun.

The **Exp. Design** menu choice offers the Analyst/Instructor a selection of analysis options, then generates appropriate run parameters. For the Kanban FATE, the options are tailored to simulation analysis and limited to the selection of a single measure of effectiveness (MoE). After selecting the MoE, the Analyst/Instructor then selects a time period for the MoE, the desired estimate precision, and a run time limit.

The **Simulation** menu choice creates the SIMAN experimental file that will drive the generic SIMAN model for this application. Because user inputs are stored in a database, any simulation/animation product that supports textual model definition can be used. Therefore, analysts/instructors and users/students would not know if and when one simulation product was replaced by another.

The raw data processor referenced in section 2.1 of this paper, while not implemented in this FATE, would be included in this section as a menu option. Functionality, it would be similar to the Calculations option.

The **Analysis** menu choice will cause the Inputs form to be shown where the User/Student specifies input values (if any were specified), and then executes the analysis. No animation is presented to the user in analysis mode.

The **Outputs** menu choice presents a tailored output report based on the most recent analysis results.

3.4 Training

Like the Analysis component, the Training component is divided into two distinct halves. The first half deals with the development of a lesson (developer). The second half deals with exercising defined lessons (student).

The **Content** menu choice allows the developer to define the lesson. This definition instructs the student on what is expected, the system description, and playing options.

The **Options** menu choice is similar to the Analysis Input choice. In this section, the developer defines the data options which will be presented to the student when exercising the lesson. The developer can specify range restrictions to further limit student options.

The **Lesson** menu choice is the first option a student will see when viewing this menu. When selected, the user will be presented with the options for the data specified by the Instructor. Once specified, the simulation model is run with animation. At the conclusion of the run, the student is presented with the results.

The **Results** menu choice allows the student to recall the most recent results. Rudimentary advice screens are also included based on a simple evaluation of the results.

3.5 Administration

In general, Administration provides a security and interface tailoring function. Users are "registered" by an Administrator as one of nine user types. The menus and menu choices available are user type dependent. For example, as shown in Figure 3, a Student will have access only to the Open and Save choices under the File menu, the Lesson and Results choices under the Training menu, and the Scorecard choice under the Administration menu. All other menus and menu choices will not appear in the user interface because they are not relevant to a Student's needs.

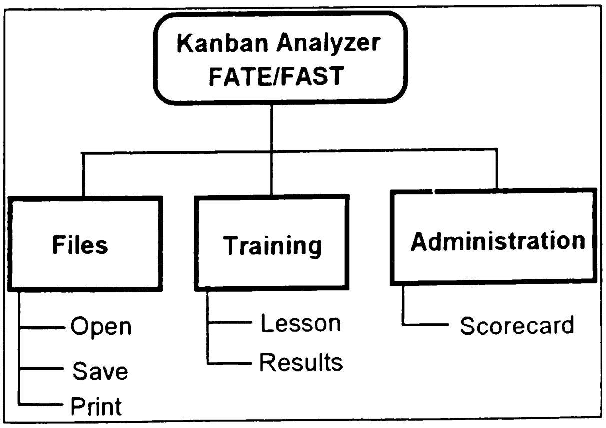

Figure 3: Kanban Analyzer Student Interface

Beyond the security and tailoring function, this section is designed to support training activities. The Users option tracks students, their scores, the number of attempts, and the date last used. It is only available to the instructor and administrator. The students have a Scorecard option they can select to view the information contained in the Users database. It reports the same information as the Users option, but only for the specific student.

4 Future Directions

Rather than a single idea, the FATE concept is a philosophy that encompasses a number of new ideas and approaches. In areas such as data input, it offers immediate benefits, while in areas such as statistical support it currently offers only future promise. The current research is centered on three areas:

1. Defining a FATE development tool kit.
2. Refining the FATE training structure.
3. Exploring the limits of what statistical assistance and safeguards can be provided.

The two prototype FATEs (Kanban Analyzer and Line Balancing Decision Trainer) produced to date have centered on simulation as the analysis tool and represent a subclass of FATEs called Focused Application Simulator & Trainers (FAST); however, the long term goal will be to provide a problem-solving delivery system that will allow analysts to move away from a specific analysis tool yet keep a standard interface structure. Because the data is contained in a database, the user will simply select a different option from the Analysis menu to move from one tool to another. One research topic will be to determine the relationship between the data required for different tools, and whether a unified storage scheme can be devised.

It is hoped that simulation vendors will recognize the need to allow for more customizable interfaces to be built. As a minimum, all vendor products should support external interfaces for input just as many already support the exporting of output data to spreadsheets and statistical software packages. Additionally, whereas constructing a data-driven simulation was straight-forward, a data-driven animation presented significant problems [Benjamin, Mazziotti and Armstrong, 1994]. A worthy growth area for any vendor offering animation software would be to allow access to animation data at run time.

5 Conclusions

If the current trend of putting simulation capability into the hands of untrained users continues, FATEs represent one evolutionary path that holds great promise. The amount of focus required by a FATE will be based on four factors:

1. Specific, unambiguous, and familiar terminology to all users
2. Simple and obvious user interface
3. The flexibility required to represent the range of systems
4. The limited set of analyses of potential interest

As presented in this paper, the ability to customize model construction and data input when the application is narrowly focused allows a much larger set of people to use the application effectively. However, the greatest gains from narrow focus applications for simulation users have yet to be realized. A detail rarely discussed in the simulation community is the fact that although vendors have made it easy to construct simulation models, little has been done to aid statistical analyses using these models. One reason for this omission is that providing such assistance requires specific knowledge of the analyses of interest and the application context. If an application is sufficiently focused, the set of analyses of interest is small and known, which allows statistical support to be incorporated into the application. If statistical support can be responsibly provided, it will represent a breakthrough accomplishment and simulation milestone.

Where and how far the simulation-based training capabilities offered by a FATE can be taken is another exciting area. For example, a goal of the Kanban Analyzer will be to help motivate students to learn by visually bringing actual industry case studies into the classroom to tie theory to practice. If the delivery process can be packaged in a "video game" style, learning will be fun.

Beyond simulation, FATEs may offer an easy delivery system to mainstream other analytical tools, particularly optimization algorithms that are used much less frequently than simulation in industry. In the other extreme, a FATE interface may help foster acceptance of simple spreadsheets when they satisfy the analytical need and reduce excessive analysis.

6 REFERENCES

Armstrong, F. Bradley. 1994. FATE Kanban Analyzer. *Unpublished NCSU Master's Project.*

Benjamin, D. Michelle, Barbara Werner Mazziotti, and F. Bradley Armstrong. 1994. Issues and Requirements for Building a Generic Animation. *1994 Winter Simulation Conference Proceedings.*

Mazziotti, Barbara Werner, F. Bradley Armstrong, and Kenneth A. Powell, Jr. 1993. Teaching Production Line Balancing with an Interactive, Simulation-based Training System. *1993 Winter Simulation Conference Proceedings*, pg. 1392-1393.

AUTHOR BIOGRAPHIES

F. BRADLEY ARMSTRONG is a Fellow Engineer for Asea, Brown, Boveri in the Engineering and Manufacturing Productivity Center where he is an internal consultant for simulation and world class manufacturing methods. He also is the owner of Simulation Engineering Associates (SEA). Prior to forming his own company, he was a Senior Staff Engineer at Hughes Aircraft Company where he worked as an internal simulation consultant. He also worked as a Senior Systems Analyst for Pritsker and Associates, and as an Operations Analyst for General Dynamics. He received a B.S. in Mechanical Engineering from the University of Texas at Austin in 1981, an M.S. degree in Industrial Engineering from Purdue University in 1986, and is completing an M.S. degree in Integrated Manufacturing and Systems Engineering. He is a professionally registered engineer in North Carolina, California, and Indiana. He is a member of **IIE** and **SCS**. His current interest is in using simulation, statistics, and GUI tools to build Focused Analysis and Training Environments.

BARBARA WERNER MAZZIOTTI is currently the Manager of Simulation Services at the Textile/Clothing Technology Corporation. For the past 4 years, Mrs. Mazziotti has been challenged with bringing the technology of process simulation to the apparel and textile industries. In this position she has focused on creating data-driven, flexible simulation models and animations for non-expert users. In addition to conducting seminars on simulation and production systems, she has completed projects with more than 20 retail, apparel and textile companies. Prior to $[TC]^2$, Barbara was on the consulting staff at Systems Modeling Corporation, teaching SIMAN and CINEMA, managing projects and doing analysis for a variety of industries and applications. Barbara began her career at General Motors as a Simulation Project Engineer for a group of metal stamping and automotive assembly plants. Barbara has a B.S. in Operations Research and Industrial Engineering from Cornell University and is completing an M.S. in Industrial Engineering at North Carolina State

University. She is a member of **SCS** and a senior member of **IIE**.

D. MICHELLE BENJAMIN is currently a Simulation Analyst at the Textile/Clothing Technology Corporation (TC^2). During the past year at TC^2 , she has completed consulting projects that address issues in warehousing and production assembly systems. She has helped in the development of flexible, data-driven simulation models and animations. In addition, she has taught seminars about simulation and its use in textile and apparel industries. She received her B.S. in Industrial Engineering and M.S. in Industrial Engineering and Operations Research from Pennsylvania State University. She is a member of **IIE** and **ASQC**. Her interests include user interfaces, generic modeling, and output analysis.

KENNETH A. POWELL, JR. is owner and operator of Simulation Solutions, a company specializing in custom simulation software. He received his BSIE and MSIE from North Carolina State University. His interests include simulation, object-oriented programming, and Microsoft Windows GUI design.