1 Identification and Significance of the Problem or Opportunity

This proposal for the Army SBIR program will significantly advance the state-of-the-art in distributed, component-based "virtual" design, construction, and simulation software tools for complete engineering systems or sub-systems.

1.1 Problem Definition

The development of new engineering systems in both commercial and combat vehicles currently involves many potentially costly steps such as hardware prototyping, where the costs are ultimately unrecoverable.

Currently there are numerous commercial Computer Aided Engineering (CAE) products that serve a useful purpose but are limited in flexibility. In large measure, the inflexibility is due to one or more of the following: 1) they tend to be "domain specific", 2) no integration across domain models, and 3) no interoperability across a network. Flexibility can be facilitated through the following: 1) use of standardized component architecture, 2) standardized data format, 3) standardized transfer protocols, and 4) distributed computing. The ultimate goal of this proposal addresses the current limitations in CAE products through a synthesis of related innovations in software engineering that have gained increasing industry acceptance.

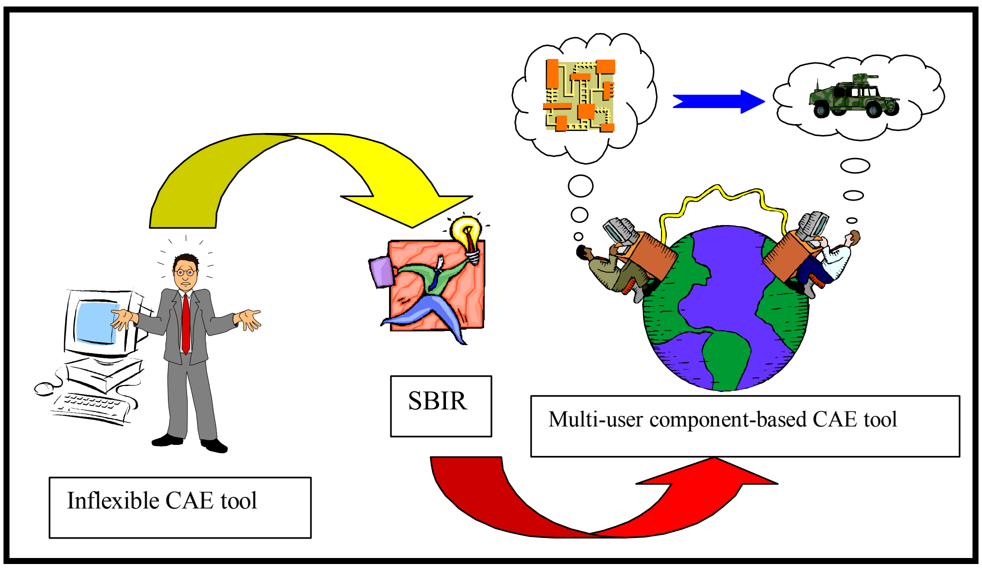

Figure 1 Comparison of current and proposed CAE tool

The key difficulty in achieving flexibility in any software system hinges on the degree of standardization that exists among the sub-systems that comprise a software system. Greater flexibility in a system is achieved when sub-system behavior and interoperability are standardized but not overly constrained. A robust, component-based standard is a particularly attractive candidate for the present inadequacy. The only international component-based standard available today is the Common Object Request Broker Architecture (CORBA) Component Model (CCM) sponsored by the Object Management Group (OMG) [1].

Over the course of the past several years we have developed one of the first and most complete software implementations of the CCM specification in the world, named Enterprise Java

CORBA Component Model (EJCCM), that is available as open source to the general public. The conceptual idea of the EJCCM resulted from work developed under the Navy SBIR program (N00014-96-C-2525) that ultimately produced a functional GUI prototype providing a drag-n-drop construction of component assemblies. The EJCCM is currently being used by CPI and other international organizations to develop innovative component-based applications for customers including the National Aeronautics and Space Administration (NASA) and the Missile Defense Agency (MDA). The estimated total government funding devoted towards the development of EJCCM is in the neighborhood of $1 million.

During the Phase I program we propose to test the feasibility of an innovative design for a flexible system engineering integration tool for virtual prototyping and simulation using the language and functionality offered by the CCM. Given the expected success of a Phase I prototype Graphical User Interface (GUI) we anticipate a substantial advancement in computer aided engineering and, ultimately, a highly flexible tool that meets the needs of both commercial and military applications.

1.2 Proposed Approach

We propose using the Rational Unified Process (RUP) [4], in conjunction with the standard Unified Modeling Language (UML) [5] to create the Virtual System Integration Lab (VSIL) design. Our software process will be consistent with the requirements of the Capability Maturity Model (CMM) Level 3 [6]. These processes and standards provide a proven framework for our organization to maximize communication and value in the design. We plan to use tools for GUI design such as Forte, and UML Modeling such as MagicDraw.

We also propose leveraging two technologies that Computational Physics, Inc. (CPI) has become intimately familiar with: an implementation of the OMG CCM specification, and a prototype GUI drag-and-drop component tool. These technologies are not only ideal for this type of application, but they also provide numerous services and specifications that free engineering resources to focus on the domain problem. Other technologies that may be useful include Defense Modeling and Simulation Office (DMSO) High Level Architecture (HLA), Extensible Markup Language (XML), and software patterns. We will be utilizing consultants from the University of Virginia (domain experts in the engineering, CAE, and Modeling & Simulation (M&S) fields) for guidance on existing methods, tools, and models.

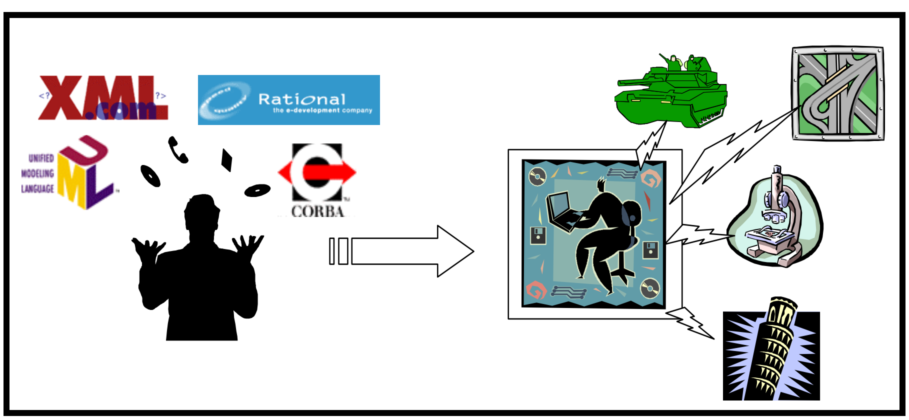

Figure 2 Synthesis of technologies to form useful system

1.3 Technical Background

The relatively recent ubiquity of computer hardware, the World Wide Web (WWW), and the potential for "pervasive computing" have accelerated the development of both the software technologies and the refinement of mathematical models to either smaller more elegant solutions, or to more massive algorithms that must run on physically large machines. However the requirements of today's software customers dictate "faster, better, cheaper." The realization of these requirements must necessarily require refinement of processes, abstraction of domain knowledge, and compartmentalization of software projects into reusable entities.

Software (and algorithm) reuse has been facilitated through the creation of object-oriented paradigms, patterns, common interfaces, standards for communications, data storage, and program management. The creation of the CORBA refines the interprocess communication requirements and allows processes on various platforms, written in various languages, to become part of a larger system. Though CORBA introduces "services" and formalizes the definition of objects using the Interface Definition Language (IDL), there still remains significant developer demands in terms of effort and knowledge.

The introduction of the CORBA Component Model begins to solve the problems of developer effort mentioned earlier, as well as provide other secondary benefits such as robust component management. Component-based technologies have gained increasing attention in the last decade [7,8]. Component-based technology breakthroughs such as Enterprise Java Beans (EJB), Distributed Component Object Model (DCOM), and especially the CCM have solidified the potential for code reuse, web-based systems deployment, and distributed user collaboration. In this section, we will demonstrate how these possibilities will be exploited to advance the capabilities of CAE tools.

The development of a CAE tool using a standardized, component-based technology such as the CCM has numerous advantages over non-standard, proprietary mechanisms frequently employed in the software industry [2,3]. These advantages include the following: 1) fully documented Application Programmer Interfaces (APIs), 2) APIs address problem areas of designing and implementing distributed systems such as component interactions, packaging, installation, deployment, persistence, events, and security, and 3) the CCM combines these services into a cohesive set of API's (still compatible with non-CCM CORBA objects) that ultimately simplify the development and administration of the software system.

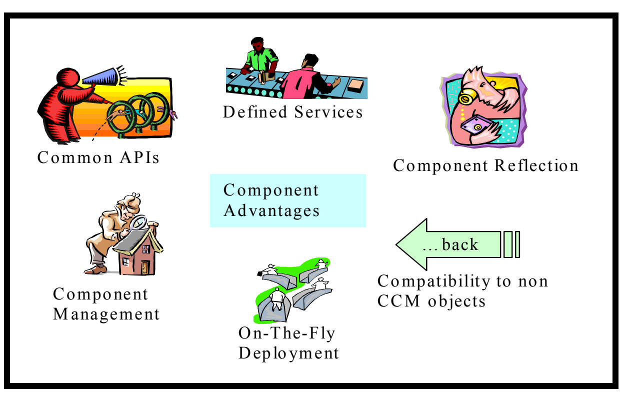

Figure 3 CORBA Component Model Advantages

1.4 Three-Phase Program Structure

Phase I work will begin by refining and following the RUP, and applying UML to capture the explicit functionality and system requirements of the itemized components. The requirements will be captured in the form of UML Use Case diagrams and Use Case Specifications that document the specific details of each end-user goal of the system. Once the requirements are established we will then use a modeling tool such as MagicDraw to develop static UML class diagrams as well as dynamic UML interaction diagrams. These UML diagrams along with class diagrams, will comprise the visual elements of the design. Textual documentation will elaborate on the UML elements and capture design elements not expressed as diagrams. With available resources, we will be in a position to capitalize on our existing drag-n-drop GUI prototype software to develop a prototype tool that will enable users to graphically configure, install, deploy, and execute assemblies of distributed electrical system components within one or more instances of EJCCM application servers.

With a successful Phase I program we will proceed to elaborate (in Phase II) on the design and begin to develop the software implementations for both the hardware and the software components of the electrical sub-system. During the Phase II effort we expect to tackle the problem of component data format translation by adopting compliance with widely adopted industry standards. We will also place considerable emphasis on extended design and functionality of a GUI that will be implemented to support collaboration between remote users. In additional to user collaboration we will develop GUI tools for rapid development of additional sub-system components. The desirable end result of the Phase II effort will be to move beyond the exemplar case of the electrical sub-system and support the development, assembly, and simulation of additional sub-systems under the guidance of the Program Manager.

Ultimately, we aim to carry the proposed program through to its fruition and develop a commercial software package that overcomes the limitations of current CAE packages. During Phase II, we will develop a coalition of potential end users of this technology from the defense and the commercial sectors. We will also develop partnering with these coalition members for more comprehensive set of specifications, testing of the tools already developed, and contracting for specific applications of interest to these partners.

2 Phase I Technical Objectives

The Phase I objective is to prepare a system design and software development plan for implementing the VSIL. The design and development plan will be based on investigations into the technical issues involved with capturing a Light Armored Vehicle (LAV) electrical system in software and executing a simulation. In addition to design, a prototype will be developed using existing in-house software products, including a component-based architecture (such as CPI's EJCCM) and a drag-and-drop component-linking GUI.

The design of such a system involves many issues and trade-offs including but not limited to the following:

1) Communications with the customer to establish user scenarios, pre-existing/implied requirements, expectations of performance and deployment, and the targeted audience of the software;

2) Establishing requirements and priorities for the system that focus effort and work into a process of finite scope and duration;
3) Consideration of the existing software within the industry:
    a. What other packages do and do not do well;
    b. Interfacing with the other packages in some way;
    c. Presenting and storing data in some standard format.
4) Evaluation of software languages, system platforms, and internet communications protocols;
5) Development of strategies for combining model elements (ie. LAV electrical system components), analysis models (ie. math model), GUI, object distribution, error handling, data formats, fidelity, etc into an extensible software.
6) Evaluation of available technologies that perform some of the system functionality (Commercial off-the-shelf (COTS) products);
7) Evaluation of LAV electrical system components (HW & SW) and how to characterize them.

As Phase I is the starting point in a longer software development process for the VSIL, Computational Physics, Inc. proposes usage of the RUP within the greater scope of a development environment consistent with CCM Level 3 (Capability Maturity Model). This will also be defined in the development plan.

Table 1 Objectives of the Phase I Project

Task	Objective
1) Use-Cases and Requirements	Define the scope of the design and what it would allow the user to do. (1-3 above)
2) System Architecture Design	Create overall system architecture including inter-process communications, GUI roles/responsibilities, domain/model decoupling (4-6 above)
3) LAV Electrical System Parameterization	Evaluate LAV Electrical Systems components and how they are characterized with respect to analysis strategies (7 above)
4) Design LAV Electrical System Model	Develop component design that comprises the software system model, and how it is analyzed/simulated. (7 above)
5) Define Development Process	Provide guidelines and processes for the Phase II development effort incorporating RUP and CCM Level 3 practices
6) Prototype Implementation	Develop GUI/Architecture prototype using existing software components, and implement basic domain electrical system & analysis component
7) Final Report	Provide design documentation and software development plan for implementing the design.

2.1 Use-Cases And Requirements

This task results in the definition of the functionality that will be built into the design. This will require communications with the customer in terms of defining what is intended as the ultimate goal of the VSIL and what types of software elements may be desired. This phase also involves proposals by CPI as to technologies, strategies, and processes that may circumvent later risks in the development process.

2.2 System Architecture Design

With requirements from the previous task, a design will be developed that satisfies the requirements, is testable, and is extensible. The design will allow refinement and prioritization of the implementation by decoupling architecture components.

2.3 LAV Electrical System Parameterization

Information regarding the LAV electrical system is collected and analyzed for data types, behaviors, abstraction, existing models and modeling limitations, and industry standards. The previous task was domain-invariant; therefore this task goes the next step in considering the specific domain problem and how the design allows for the various data/functionality for each electrical system component. The analysis strategy for such a system is also determined in this task depending on requirements and/or existing models.

2.4 Design LAV Electrical System Model

In this task the actual LAV system is converted to a component design and analysis tool design. The specifics of how each component is characterized in software (ie. object methods and attributes), altered (data types and allowed values), and analyzed will be defined. This task is considered a domain-dependent extension of the design created in Task 2.

2.5 Define Development Process

Phase I is considered the first step in the proposed Rational Unified Process of implementing the VSIL. The Rational Unified Process defines an iterative software development process. Software is designed, implemented, and tested in stages, each stage having similar activities, with the software growing in complexity. This task will define those activities and propose a schedule for Phase II based on the requirements and resources. Consistent with CMM Level 3 processes, we will propose the artifacts that will come out of the rest of the phases of this SBIR – several artifacts will have already been created by the time of conclusion of this task.

2.6 Prototype Implementation

A prototype will be implemented using existing in-house software products. CPI is currently the host of an implementation of the CORBA Component Model called the EJCCM. The EJCCM is well suited for this project due to its distributed, robust, industry standard treatment of components and the issues associated with such architectures.

2.7 Final Report

This task results in a final report summarizing the design, trade-offs, and decisions of Phase I. Feasibility for implementing the design in Phase II given particular resources will be discussed, and strategies of risk-reduction/ prioritization will also be presented. Included with the final report will be the design and development plan documentation.

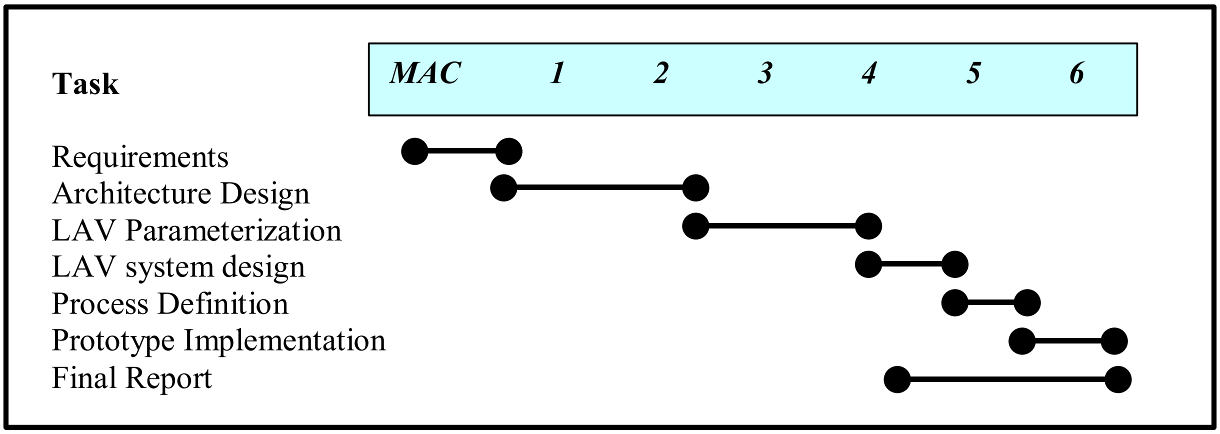

Figure 4 Proposed Time-line for Phase I

3 Phase I Work Plan

The following work plan details the flow of tasks to accomplish Phase I objectives. The tasks are designed to provide increasing detail and functionality to the artifacts that result from Phase I.

3.1 Use-Cases And Requirements.

This task can be divided into 3 subtasks: Use-cases, requirements, and test plan. The development of Use-Cases will involve communications with the customer and discussions regarding expectations, L&F (look and feel), system deployment/delivery, system options (ie. configuration and command-line options), and importable types (ie. AutoCAD files.)

Requirements are created based on customer discussions, reasonable system behavior/restrictions, available technologies, schedule, available resources, and technical trade-offs. Requirements define the set of functionality and measurable characteristics of the system that constitute a successful satisfaction of the objectives. Requirements can be added-to later, but this task outlines the minimum set of Phase II requirements.

Requirements fall into the following categories:

1) Implementation Requirements – These include supported operating system(s), software language(s) support, delivery procedures, documentation, help-desk support, system build procedures, etc.
2) Architecture Requirements – These include data/object/interface distribution, error handling, reliability issues, persistence facilities, threading policies, etc.
3) Administration Requirements – These include configuration, administrator interactions, security.
4) Usage Requirements – These include all user interactions with the system, user guide, menus, windows, dialogs, L&F;
5) Execution Requirements – These include performance, analysis, domain extensibility, data validation, testing methodology, Hardware-In-The-Loop (HITL).

Each of the requirements for the above categories will be elaborated and documented. A test plan will be developed that outlines the set of functionality tests for each of the requirements groups outlined above. At this time a build procedure will be defined as this will greatly enhance the ease of testing (unit, system, and regression.) The test plan will be divided into three separate sections: 1) Infrastructure, 2) GUI, 3) Domain Validation.

Table 2 Work Plan for Task 1

Subtask	Objective
1) Develop Use-Cases	Define scenarios for user activities, and abstract behaviors of the system
2) Develop Infra-structure Requirements	Define requirements for the categories detailed above.
3) Develop Test Plan	Document high-level functional testing procedures for validation of objective from the standpoint of requirements

3.2 System Architecture Design

The system architecture will be compartmentalized based on the famous Model-View-Controller (MVC) [9] pattern that originated within the SmallTalk software community. The main advantage of approaching the system architecture in this fashion (from the point of view of dependencies and roles) is the clear definition of the roles of each object. Objects whose roles cross the MVC boundaries can be further divided so that the divided objects have clearer roles.

Once the roles and responsibilities are defined, the interfaces (behaviors and data types) between the objects can be refined. The control elements of the system include the component framework. At this time we propose usage of the Enterprise Java CORBA Component Model developed by CPI. The EJCCM already provides many facilities: system infrastructure, machine-independent native data types, error handling, data persistence/object lifetimes, and process/data distribution. The EJCCM is well suited for robust component management and the handling of many complicated behind-the-scenes issues in order to simplify component design and implementation. We propose incorporating EJCCM in its totality, thus freeing valuable resources for other high-risk areas of the implementation. We will provide high-level design and usage documentation along with EJCCM.

The GUI itself will be created with a GUI tool (ie. Forte For Java.) This process involves designing menus, mouse click options (ie. right mouse click brings down menu? What's on the menu?), scenario flow (ie. which windows come up based on where the user is in the workflow), and dialog categorization. Computational Physics, Inc. has developed a program called "Mega-Tool" – a drag-and-drop prototype GUI for graphically connecting and manipulating components within a component architecture. In Phase I we will propose reusing most of this design, and using most of the component-manipulation code in Phase II.

The next subtask is to map GUI actions to object method call sequences, and error handling. Some GUI actions will activate distributed components; search for library components, run the analysis tool, add components, or shutdown the entire application. This is a direct translation from use-cases to software method sequences.

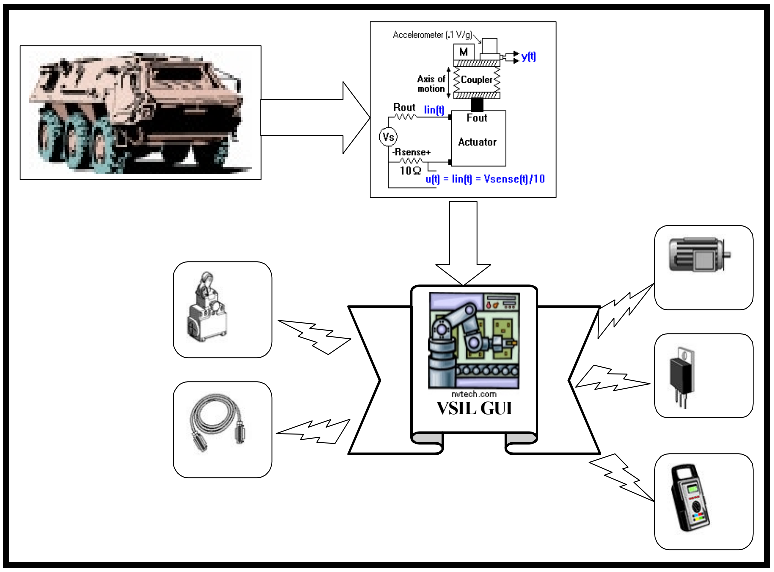

Figure 5 Progression from LAV to domain-extraction to VSIL Software System

The design will be extended to include Discrete Event Simulation (DEVS)-specific components. The design decisions that are involved here include current technologies, foreseeable requirements not-yet-defined for Phase II, high-priority functionality, high-risk functionality, and availability of canned software products that can be incorporated "easily" into the current infrastructure design. The design will also reflect the impact of various domains (electrical systems, mechanics, hydraulics, etc) on the "general system component" interface in terms of reflective mechanisms so that components can discover about each other for the purposes of Input/Output (I/O) and connection validation. This subtask will also involve consultations with Prof. Pradip Sheth and Prof. Ron Williams of the University of Virginia.

A proposal for incorporating HITL (hardware in the loop) into the VSIL will be made, and may include EJCCM Components that access hardware directly, or EJCCM Components that are wrappers of some other technology (ie. different languages such as C/C++, or different protocol, such as HLA [10])

Table 3 Work Plan for Task 2

Subtask	Objective
1) Define Infrastructure Component Framework	Define application-level functionality – object distribution mechanism, error/message handling, test-assisting facilities, generic system IO/logging, object persistence, etc.
2) Design visual GUI layout	Develop GUI using GUI design tool (ie. Forte) –define windows, dialogs, menus, etc.
3) Design GUI<->Infra-structure Interactions/ behaviors	Design GUI behaviors and how the GUI has access to- and responds to the distributed component infrastructure
4) Design Domain-Extensible simulation functionality	Extend infrastructure to include M&S functionality, time management – ie. DEVS, real-time, etc.

3.3 LAV Electrical System Parameterization

This task will involve communications with the customer and document exchange in order to outline the exact specification of the components and the intended level of detail for the Phase II implementation work. Tasks 3&4 will be the focus of the consultations with Prof. Pradip Sheth and Prof. Ron Williams of the University of Virginia.

Computational Physics, Inc. will need to obtain LAV documentation concerning the LAV electrical subsystems (CPI is capable of dealing with the various Department of Defense (DoD) security issues as appropriate.) The documentation will serve as rules for parameterization of the domain elements. The parameterization will involve the following considerations:

Domain application: Does the customer include the element as a part of the "electrical system?" Is the element a mechanical, electrical, or software component (some elements may have mechanical and electrical properties – the electrical properties are of importance in this case)?

Element Properties: What measurable properties exist for the element? What is its I/O? Is the IO discrete or continuous? What are the ranges of the properties (ie. legal values, 0-5V for example)? What are reasonable "default values" for the properties (ie. lights being 12V)?

Element Constraints: Does element control others indirectly or directly? What constraints are placed on the element? Can implicit rules be applied to the element? (Ie. battery voltage based on drain rate)

Analysis: What system-wide parameters change over time? What external stimuli are applied to the system? What defines the end of the analysis run? Is the system linear or non-linear? What models exist for special cases of this system/subsystem?

An analysis strategy will be designed that represents a reasonable scope of effort, known models/tools/CAE I/O, and added value to the customer [11]. An attempt will be made to abstract the analysis strategy where possible, extending the ideas to non-electrical domains, such as dynamic mechanical systems. The analysis strategy is not the analysis component itself, but represents some of the possible configurations, ie. DEVS, domains, and inputs and outputs. The electrical system component model and analysis component designs will conform to the overall analysis strategy, yet be a subset of the possibilities.

Table 4 Work Plan for Task 3

Subtask	Objective
1) Obtain LAV documentation	Obtain documents, communicate with LAV engineers, and document relevant findings
2) Extract LAV components and parameters	Categorize each system component, assign potential I/O / behaviors
3) Determine Analysis Strategy	Generalize analysis procedure for the system given requirements from Task 1

3.4 Design LAV Electrical System Model

This task converts the information gathered in Task 3 into a domain model of the LAV electrical system. This will include physical, software, and analysis elements. Tasks 3&4 will be the focus of the consultations with Prof. Pradip Sheth and Prof. Ron Williams of the University of Virginia. The interconnections and dependencies of the LAV electrical system will be translated into data, event, and behaviors in the software model. This includes data that is shared between elements, timing issues, events and milestones within the functioning system, algorithms governing the software elements, software I/O data and behaviors, and physical & electrical & other constraints.

Given the model elements, interfaces (ie. code) for the software components will be defined. Dependencies will be realized via shared data types and events. The result of this activity will be a complete description of the software components that (when implemented) need only interconnection via the GUI.

The Analysis Component (AC) will likely have a simple interface (ie. it will accept other components, and have a "runnable" functionality.) As elements are added to an AC through drag-and-drop operations, the AC will have the opportunity to reject inputs, based on the capabilities of the AC. These exceptions (indicating an attempt by the user to analyze a situation that is too complex for the AC) will be propagated to the GUI as "can't do that" messages. (This scenario serves as one example of how the GUI and component architecture work together – despite the strong decoupling between them.) The intent is to be able to provide any AC to the GUI (via EJCCM deployment mechanism), and be able to connect components to it, as long as the AC is able to handle them. As more complex AC's are developed, they would naturally allow more complex inputs/relationships between the components.

Table 5 Work Plan for Task 4

Subtask	Objective
1) Define System Component Behaviors/ Messages/ Events	Define dependencies, interactions, and analysis needs for each component and how they relate to the other components in the system
2) Define System Component Interfaces	Translate subtask 1 elements to full component interface definitions
3) Design Analysis Component	Define analysis component that interfaces with domain components (data) and GUI (output/events during execution)

3.5 Define Development Process

Task 5 involves tailoring the RUP and CMM Level 3 processes to CPI, the customer, and this SBIR. This process involves consideration of the schedule requirements, documentation requirements, resources, and available technologies/COTS products. This task will create a preliminary schedule for Phase II, define the build and iteration sequences, and define the pertinent artifacts (documentation, meetings, software, etc) that result from each iteration.

The Phase II functionality will be divided into a series of milestones with assigned durations, tests, and acceptance criteria. A Configuration Management (CM) plan will be created such that risks are minimized and the method for the tracking of errors is defined. A deliverable will be assigned at the end of each iteration. A deliverable may be documentation, software, and/or test results.

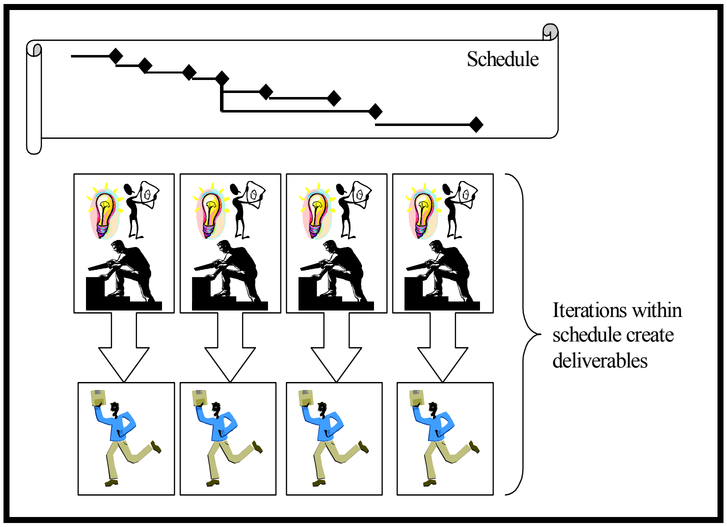

Figure 6 Progression From Schedule Definition To Deliverables

Table 6 Work Plan for Task 5

Subtask	Objective
1) Taylor RUP To CPI And This SBIR Software Project.	Extract RUP elements and tailor them to fit our organization, the customer, and the schedule requirements of Phase II.
2) Document Process	Document tailored RUP for Phase II and beyond

3.6 Prototype Implementation

Task 6 will provide prototype software that allows visual connecting of electrical components, and the running of a simulation analysis component. The electrical model (a series of components like wires, actuator, and sensor, and battery connected in series) and analysis component will be basic, however the most important functionalities will be demonstrated. These functionalities include the component-based architecture, and drag-and-drop GUI that allows connecting components, and a demonstration of domain specialization of components. The prototype will implement a subset of the delivered Phase I design.

The process of developing the prototype will encompass the following subtasks: 1) Generate the GUI code using Forte For Java, 2) Insert the EJCCM-access code, and 3) Define a set of components that demonstrate the basic design and electrical analysis.

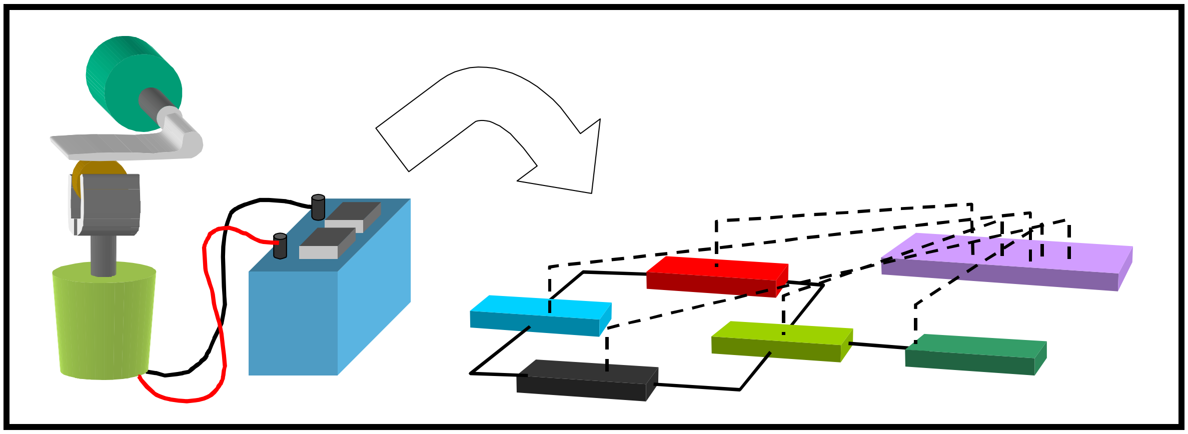

Figure 7 Translation of Simple Electrical System from Domain Entities to Components/Links

The premise for the simple electrical system example will be to observe over time (discrete time steps) the characteristics of the system: current (determined within purple element in Figure 8), voltage, and actuator position (within light green element) as the battery is drained of charge. An analysis component will also be created to interpret the characteristics of each component (the physics of the electrical system) and produce basic output.

Table 7 Work Plan for Task 6

Subtask	Objective
1) Generate GUI Skeleton Code	Using GUI generator program create basic GUI classes , dialogs, and menus, etc.
2) Populate GUI Code with EJCCM calls	Implement EJCCM access calls for locating, creating, installing, and removing components into empty GUI class methods
3) Develop Simple Domain System Components and Analysis Component	Implement components, and their instantiation within the GUI via library lookup
4) Run Simulation	Implement linking components and data sharing, implement starting of analysis component.

3.7 Final Report

During the design process of Phase I, many issues will be considered. Technical tradeoffs will be made, and requirements defined for Phase II. The Final Report will include these discussions, as well as the Requirements, Design, Development Plan, Test Plan, and Configuration Management Plan documents. The whole of the documents will be reviewed.

The feasibility of implementing the design and development plan within Phase II will be discussed, including risks, technical challenges, schedule, resources, etc. Finally a discussion of the prototype will present a review of the functionality implemented, the issues encountered, and the changes to the design that may be foreseen for Phase II, if any.

Table 8 Work Plan for Task 7

Subtask	Objective

1) Review Design and Process	Document overview of design and software process.
2) Evaluate Feasibility For Phase II	Document proposed Phase II schedule and discuss feasibility of schedule
3) Present prototype results	Discuss the implementation of the prototype and issues, and Phase II considerations.

4 Related Work

Work for the MDA, titled the Battlespace Environment Signatures Toolkit (BEST), involves an innovative component-base reengineering of legacy simulation software. The BEST system, which is built upon the EJCCM framework, will enable users to construct and deploy assemblies of distributed battlespace components (hardbodies, terrain, clouds, missile plumes, celestial bodies) that participate in distributed simulations and High Level Architecture (HLA) federations.

Battlespace Environments and Signatures Toolkit (BEST): This Missile Defense Agency Program is dedicated to the design and development of the next generation high fidelity simulation software that encompasses all of the battlespace entities that are detectable by ground or space based sensors. The BEST software system is built on top of CPI's EJCCM implementation. In addition to further improvement of EJCCM, CPI is responsible for requirements management and the design and implementation of simulation engines for BEST.

Photochemistry Phenomenology Modeling Tool (PPMT) & Transport Phenomenology Modeling Tool (TPMT):: These two NASA projects are sponsored by the Applied Information Systems Research Program (AISRP) and are dedicated to the design and implementation of the next generation first principles modeling codes for conducting research in planetary atmospheres. The software currently being developed for these two projects are built on CPI's EJCCM implementation.

Component-based Technology: CPI is an influencing member of the Object Management Group and is currently participating in the Revision Task Force for the CORBA Component Model specification. CPI's current involvement with the CCM RTF includes assistance in revision and defect removal from the CCM specification and participation in presentations and tutorials at OMG technical meetings.

5 Relationship with Future Research and Development

The result of this Phase I work will positively impact the computer aided engineering technology development as a whole. This work will have many ramifications: 1) solidification of standards in the CCM and domain industries involved, 2) additional design and abstraction of the M&S domain, 3) development of model integration strategies, 4) possible performance benchmarking for future projects, and 5) VSIL, a tool that is capable of integrating HITL, component libraries, GUI drag-and-drop, and deployable models into a cohesive CORBA Component application. The standards-driven component-based approach being proposed in this Phase I will have far reaching applications for both the military and commercial applications. This work will be a step towards ensuring the leading role of the United States in this important technology.

6 Potential Post Applications

The commercial value of CAE tools has grown steadily as these tools have been applied to reduce the time-to-market and overall cost of introducing new engineering systems. The effectiveness of these CAE tools has been limited by both computer hardware and software technologies. The exponential growth of the Internet has spurred similar growths in computer hardware and networking to a point that has outpaced advances in software. Although software tools are increasingly installed on multiprocessor computers the software tools are typically not designed to utilize the available resources. Through advances in pervasive computing on heterogeneous platforms offered by CORBA and the well-defined flexibility of the standardized CORBA Component Model future software packages can seamlessly utilize all available computing resources allowing for significant improvements in functionality and performance.

The system proposed in this SBIR will also be capable of hosting virtually any type of M&S activity, whether battlefield sims, weather predictions, economics, land management, etc. The limits will be imposed by the component libraries and analysis models themselves – and updates to them do not require VSIL code modifications.

7 Key Personnel

In this work for the Army CPI's experts in the field of software engineering will join with experts in the fields of mechanical engineering and electrical engineering to create a team that is dedicated to achieving a successful Phase I program and beyond. Mr. Sean Parker of CPI will lead the project as principal investigator and will be supported by Mr. J. Scott Evans of CPI, Professor Pradip Sheth of the University of Virginia and Professor Ron Williams also of the University of Virginia.

Sean Parker is a Manager of Information Technology for Computational Physics, Inc. and will be the Principal Investigator (PI) for this program. Mr. Parker has the following educational background:

BS	Physics	George Mason University	1993

Qualification Summary: Mr. Parker spent the first 4 years with CPI as the Naval Research Laboratory Special Sensor Ultraviolet Limb Imager (SSULI) Ultraviolet Calibration Facility Operations Manager. This task involved programming mechanical systems involving interaction of the PC data busses including VME standard, RS-232-C, GPIB, and fiber optics. Mr. Parker also became familiar with electrical system maintenance, printed circuit board design, layout and assembly. Mr. Parker performed much of the calibration done in the facility, and provided raw data and interpretation using custom display software and third party software such as Microsoft Excel, and RSI's Interactive Data Language. Mr. Parker has also assisted CPI scientists in the programming of data generation/GUI codes for the US Naval Research Laboratory (NRL)/SBIR and NRL/SSULI projects. These tasks involved C/C++ for Unix and Dos-based systems, and RSI's IDL. Mr. Parker contributed to the design and implementation of the Virtual Data Center Software for the Naval Research Laboratory. Skills utilized on this project include design using Rational Rose, the Java programming language with distributed extension using CORBA, and web-based technologies such as Hypertext Markup Language (HTML) and Common Gateway Interface (CGI). Mr. Parker is currently working on two physics-based models: validating a chemistry relaxation algorithm built using EJCCM (Enterprise Java CORBA Component Model) and the BEST (Battlespace Environment Signatures Toolkit) project. Mr. Parker is also involved with the design and implementation of the EJCCM. He has considerable experience with a variety of computer platforms, associated operating systems, and programming languages including C++, Java, Interactive Data Language (IDL), Interface Definition Language (IDL), HTML, and XML.

During a prior employment with CACI, Inc., Mr. Parker served as Software Architect for a battlefield HLA simulation project (PROPHET) that simulates Emitter/TUAV signal acquisition. Technologies include Java, SmallTalk, MS Access database. Mr. Parker created project deliverables and internal documentation, reverse engineered systems into design, guided new design changes, participated in prototyping through design (Rational Rose) and coding (SmallTalk). Mr. Parker also produced review procedures, initiated software pattern reuse procedures between projects, and assisted in scheduling and resource allocation.

J. Scott Evans is the Chief Technology Officer for Computational Physics, Inc. Mr. Evans has the following educational background:

MS	Applied Physics	George Mason University	1993
BS	Mathematics	George Mason University	1989
BS	Physics	Virginia Polytechnic & State University	1988

Qualification Summary: Mr. Evans directs the development of innovative software design methodologies and software architectures that utilize new technologies in the areas of Information Technology including component architectures, object-oriented and component-based design and implementation, web-based user-interfaces, and distributed computing. Mr. Evans is the principal author the Enterprise Java CORBA Component Model software. EJCCM is one of the first and most complete software implementations of the CORBA Component Model specification in the world. Mr. Evans is currently supporting the design and implementation of the Battlespace Environment and Signatures Toolkit (BEST) sponsored by the Missile Defense Agency. Mr. Evans is also chief architect of the Photochemistry Phenomenology Modeling Tool (PPMT) and the Transport Phenomenology Modeling Tool under development for NASA's Applied Information Systems Research Program. Mr. Evans is recognized as an expert in the area of component-based software engineering and is a co-author of standard CORBA Component Model specification. Mr. Evans is currently participating in the Revision Task Force of the CCM specification with the Object Management Group. He also has considerable experience with a variety of computer platforms, associated operating systems, and programming languages including C++, Java, Interactive Data Language (IDL), Interface Definition Language (IDL), Ada 83, Ada 95, FORTRAN, PASCAL, HTML, XML, and VRML.

Pradip N. Sheth is an Associate Professor, Mechanical and Aerospace Engineering at the University of Virginia**.**
Dr. Sheth has the following educational background:

PhD	Mechanical Engineering	University of Wisconsin	1971
MS	Mechanical Engineering	University of Wisconsin	1968

BE Mechanical Engineering University, Baroda, India 1965

Qualification Summary: Dr. Pradip Sheth has been involved in the development and application of modeling and simulation, and CAE of mechanical systems for over thirty years. Prior to joining the University of Virginia in December, 1985, Dr. Sheth managed the Computer Applications and Mechanical Systems Technology programs at the Advanced Technology Center of Allis Chalmers Corporation in Milwaukee, Wisconsin from 1974-1985, and from 1971-1974 Dr. Sheth was at the University of Michigan as a Senior Research Engineer/Lecturer engaged in the development of automobile system models for vibration/shake analysis. Dr. Sheth and his collaborator Dr. John Uicker at the University of Wisconsin are known for the development of the first comprehensive and general purpose computer aided engineering system for 3-Dimensional mechanical machinery and vehicle systems. This system, introduced in 1972 and called IMP, is still in use at a number of industrial and academic institutions, and the underlying numerical algorithms have formed a basis for a variety of more recent mechanical simulation and CAE systems. At the University of Michigan, Dr. Sheth participated in an early Ford Motor Company research program for the development of vehicle simulation based on a building block approach to assembling vehicle subsystems and at that time addressing the linear dynamic behavior. At Allis Chalmers, Dr. Sheth developed CAE models and simulation software for the development of off road vehicles including farm tractors, combine harvesters, forklift trucks, construction equipment, garden tractors, and snow blowers. Since 1985, at the University of Virginia, Dr. Sheth is involved in the development of modeling/optimization tools for multibody systems and their applications to mobile robots, biomechanical movements of walking and lifting, and Mechatronic systems.

Publications:

(1) Gopalakrishna, S.V., and Sheth, P.N., "Dynamic Modeling of a Zero Turn Radius Mobile Robot", IASTED Conference Proceedings, *Applications of Control and Robotics*, January 8-10, 1996, Orlando, FL.

(2) Hodges, T.M., and Sheth, P.N., 1994, "Toward a New Modeling Paradigm for Constrained Mechanical Systems", *Journal of Mechanical Design*, Transactions of the ASME, March 1994, Vol. 116, PP.80-87.

(3) Gopalakrishna, S.V., Sheth, P.N., and Kennedy, K., "Motion Modeling and Simulation of a Robotic Vehicle", December 1992, IMACS Proceedings, *International Symposium on Mathematical Modeling and Scientific Computing*, Bangalore, India.

(4) Hughes, C.J., Weimer, W., Sheth, P.N., and Brubaker, C.E., 1992,"Biomechanics of Wheelchair Propulsion as a function of Seat Position and User-Chair Interface", Archives of Physical Medicine and Rehabilitation, March, 1992, Vol. 73, PP.263-269.

(5) Sheth, P.N., Craig, K., Mattice, M., and Banks, S., 1991, "Design and Development of a Computer Aided Engineering System for Controlled Multibody Systems", *Journal of Engineering Design*, 1991, Vol. 2, No. 3, pp. 175-195.

(6) Sheth, P.N., Hodges, T.M., and Uicker, J.J.,Jr., 1990, "Matrix Analysis Method for Direct and Multiple Contact Multibody Systems", *Journal of Mechanical Design*, Transactions of the ASME, June, 1990, Vol. 112, pp. 145-152.

(7) Claar, P.W., and Sheth, P.N., 1987, "Modal Analysis Methodology for Articulated Machinery and Vehicles", SAE Publication SP-722, 1987, *Computer Applications in Design and Manufacturing*, SAE Paper No. 871660.

(8) Sheth, P.N., and Uicker, J.J., Jr., 1972, " IMP (Integrated Mechanisms Program), A Computer Aided Design Analysis System for Mechanisms and Linkage", *Journal of*

Engineering for Industry, Transactions of the ASME, 1972, Vol. 94, Series B, No. 2, pp. 454-464.
(9) Sheth, P.N., and Uicker, J.J., Jr., 1971, "A Generalized Symbolic Notation for Mechanisms", *Journal of Engineering for Industry*, Transactions of the ASME, 1971, Vol. 93, Series B, No. 1.

Ronald D. Williams is an Associate Professor of Electrical & Computer Engineering and Associate Professor of Mechanical Engineering at the University of Virginia.
Dr. Williams has the following educational background:

PhD	Electrical Engineering	Massachusetts Institute of Technology	1984
MS	Electrical Engineering	University of Virginia	1978
BS	Electrical Engineering	University of Virginia	1977

Dr. Williams performs research in modeling and simulation and embedded computer system safety and security. He has recently constructed a cluster computer for distributed simulation of safety critical systems. His current research using this cluster computer is extending the distributed simulation to include real-time hardware-in-loop simulation capabilities. The next planned effort is to extend the hardware-in-loop capabilities to hardware-in-loop at a remote sites with connectivity established using a virtual private network over the public internet.

Dr. Williams teaching responsibilities have been primarily in the areas of computer system design, embedded computing, and computer security. He has been recognized with several awards for teaching including the All University Teaching Award from the University of Virginia.

Publications:

(1) Grant, M., & Williams, R., "Statistical Processing for Gastric Slow Wave Identification," Medical & Biological Engineering and Computing, (in press).
(2) Williams, R., Klenke, R., & Aylor, J., "Teaching Computer Design Using Virtual Prototyping," IEEE Transactions on Education, (in press).
(3) VonAncken, A., Williams, R., & Salinas, M., "A Coarse/Fine Search PN Code Acquisition Scheme," IEEE Transactions on Aerospace and Electronic Systems, v. 37, n. 1, January 2001, p. 280-285.
(4) Maslen, E., Sortore, C., Gillies, G., Williams, R., Fedigan, S., & Aimone, R., "Fault Tolerant Magnetic Bearings," *ASME Journal of Engineering for Gas Turbines and Power*, v. 121, n. 3, July 1999, pp. 504-508.
(5) Gray, M., Williams, R., and Chen, J., "A Prototype Algorithm for Automated Determination of Gastric Slow Wave Characteristics," *Medical & Biological Engineering and Computing*, v. 31, n. 1, January 2000, p. 49-55.
(6) Klenke, R., Kumar, S., Aylor, J., Johnson, B., Williams, R., and Waxman, R., "ADEPT: A Unified Environment for End-to-End System Design," *Current Issues in Electronic Modeling*, J-M Berge, ed., Kluwer Academic Publishers, 1997.
(7) Schaefer, P., Williams, R., Davis, G., and Ross, R., "Accuracy of Position Detection Using a Position Sensitive Detector," *IEEE Transactions on Instrumentation and Measurement*, v. 47, n. 4, August 1998, pp. 914-919.
(8) Knospe, C., S. Fedigan, R. Hope, and R. Williams, "A Multi-Tasking DSP Implementation of Adaptive Magnetic Bearing Control," *IEEE Transactions on Control Systems Technology*. v. 5, n. 3, March 1997, pp. 230-238.

(9) J. Aylor, R. Waxman, B. Johnson, & R. Williams, "The Integration of Performance and Functional Modeling in VHDL," chapter 2 in *Performance and Fault Modeling with VHDL*, J. Schoen, Ed., Prentice-Hall, Englewood Cliffs, NJ, 1992.

(10) Klenke, R., Kumar, S., Aylor, J., Johnson, B., Williams, R., and Waxman, R., "ADEPT: A Unified Environment for End-to-End System Design," *Current Issues in Modeling*, Kluwer Academic Publishers.

8 Facilities / Equipment

Computational Physics, Inc. plans to conduct the proposed tasks at our corporate headquarters in Springfield, Virginia. The following description provides an overview of CPI's facilities, from which the appropriate dedicated space and equipment will be made available. CPI's Springfield offices consist of over 6,000 sq. ft. of office and computer facilities immediately accessible to the I495 Beltway. From this location, CPI personnel can be at ARL, ONR, most DoD, and any one of three international airports within 40 minutes.

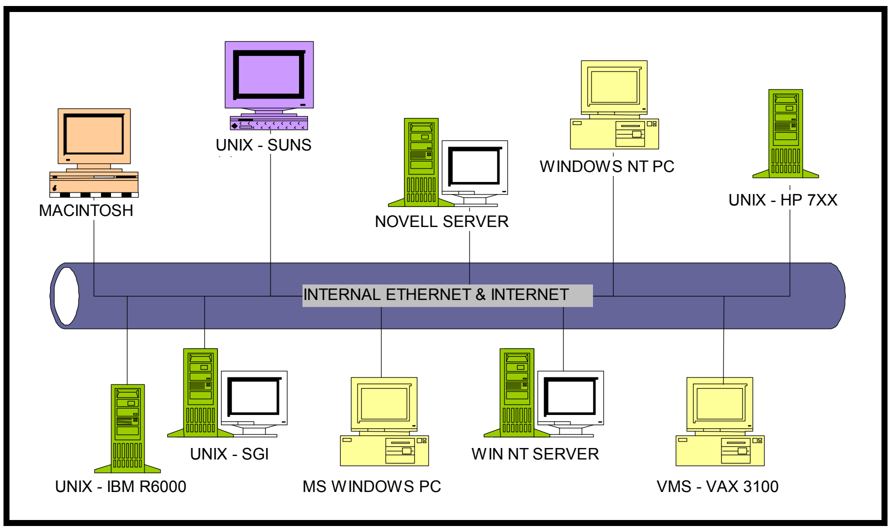

Figure 8 Schematic of CPI's Network Facilities

In addition to our physical facility, CPI has significant local and remote computing capabilities. CPI operates an in-house multi-user environment with computer resources distributed on a LAN supporting TCP/IP and IPX, which facilities are shown in Figure 8. Computational facilities are configured around Dell Linux/NT Servers and SGI/HP/PC clients. Every employee is provided either an IBM-compatible or a UNIX workstation through which they may access any corporate resource, including both UNIX and Windows software. CPI also has direct access to the Internet through NASA and modem/Internet access to array processors and supercomputers through scientific collaborations with universities and government laboratories around the country including APL, ARL (MSRC), AFRL, NRL and others.

9 Consultants

Computational Physics, Inc. will employ the services of Prof. Pradip Sheth from the University of Virginia and Prof. Ron Williams also from the University of Virginia. They will complement

CPI's technology base in the issues and implementation of domain specific CAE tools and integration of these tools. Professor Williams will provide expertise in the areas of modeling and simulation of electrical/electronic subsystems, modeling of software, and hardware-in-the-loop models. Professor Sheth will provide CAE technology related to mechanical/vehicle system modeling and issues related to integration of multiphysics subsystems across domains. Both consultants will collaborate on modeling and integration of discrete event systems with continuous time models and control logic embedded in modeled system's software.

10 Prior, Current, or Pending Support

There is no other proposal submitted by CPI in response to this solicitation that is substantially the same as this proposal, or is pending with another federal agency of DoD Component or the same DoD Component

11 Literature Citations

[1] *CORBA 3.0 New Components Chapters,* Object Management Group Standard, Nov 3, 2001, ed. Merle, Dr. P.
[2] Siksik, D.N., *STRIVE: An Open and Distributed Architecture for CGF Representation*, CAE Electronics, Ltd.
[3] Follen, G., Kim, C, Lopez, I., Sang, J., Townsend, S., *A CORBA-based Development Environment for Wrapping and Coupling Legacy Scientific Codes*, IEEE 0-7695-1296-8/01, 2001
[4] *Rational Unified Process FAQ,* Rational Software, Inc., URL: http://www.rational.com/products/rup/faq.jsp
[5] *UML Resource Page*, Object Management Group, URL: http://www.omg.org/uml/
[6] *Capability Maturity Model® for Software,* Carnegie Mellon University Software Engineering Institute, URL: http://www.sei.cmu.edu/cmm/cmm.html
[7] Gannon, D., et al., *Component Architectures for Distributed Scientific Problem Solving*, Indiana University, Dept. of Computer Science.
[8] Armstrong, R. et al., *Toward a Common Component Architecture for High-Performance Scientific Computing,* In *Proceedings of the 1999 Conference on High Performance Distributed Computing*
[9] *Model-View-Controller*, (author unknown)
URL: http://www.object-arts.com/EducationCentre/Overviews/MVC.htm
[10] *High Level Architecture*, Defense Modeling and Simulation Office, URL: https://www.dmso.mil/public/transition/hla/
[11] Benjamin, P. et al., Knowledge Based Systems, Inc., *Simulation Modeling At Multiple Levels Of Abstraction*, In *Proceedings of the 1998 Winter Simulation Conference, 391-398*.

12 Letter of Acceptance

SCHOOL OF ENGINEERING AND APPLIED SCIENCE
PREAWARD RESEARCH ADMINISTRATION
University of Virginia
122 Engineers' Way Trailer 1
P.O. Box 400257
Charlottesville, Virginia 22904-4257
Phone: 434.924.6272 Fax: 434.924.6270

August 8, 2002

E. W. Chappell, Jr.
Chief Operating Officer
Computational Physics, Inc.
8001 Braddock Road, Ste 210
Springfield, VA 22151-2110

RE: DOD/ARMY SBIR Phase I submission to Topic No. A02-244
Research and Technical Area: Information Systems

Dear Mr. Chappell:

Dr. Pradip N. Sheth, Department of Mechanical and Aerospace Engineering and Dr. Ronald D. Williams, Department of Electrical and Computer Engineering, School of Engineering and Applied Science, University of Virginia will be available to provide subcontractor support to Computational Physics, Inc. for the DOD/ARMY SBIR Phase I project entitled "Virtual System Integration Lab (VSIL) – A Flexible System Integration Tool for Virtual Prototyping & Simulation."

The budget for the University of Virginia for the 5 month project with expected inclusive dates of December 1, 2002 – April 30, 2003 is $11,400. The University of Virginia understands an additional $5,000 option is possible at a later date.

Participation of the University of Virginia is contingent upon approval of Terms and Conditions acceptable to the University and Computational Physics, Inc.

We look forward to working with Computational Physics, Inc. and anticipate a successful project.

Sincerely,

Haydn N. G. Wadley
Associate Dean for Research

Ronald D. Williams, Co-Principal Investigator

Michael G. Glasgow, Jr., Director
Office of Sponsored Programs

Pradip N. Sheth, Principal Investigator

Joseph A. C. Humphrey, Professor and Chair

HNGW:tj

Enclosures: (4) SEAS Proposal No. MAE-CPI-1257-03

c: Dr. P. N. Sheth
Dr. R. D. Williams
Dr. J. A. C. Humphrey
Dr. J. H. Aylor
Mr. M. G. Glasgow, Jr.
Ms. F. B. Cline
Ms. D. E. Van