

## **Dynamic Data Driven Application Systems**

*"Creating a dynamic and symbiotic coupling of application/simulations with measurements/experiments"*

### **Preface**

The primary objective of this workshop was to explore research opportunities leading to the creation and enablement of a new generation of dynamic/adaptive applications. The novel capabilities to be created are application simulations<sup>1</sup> that can dynamically accept and respond to "online" field data and measurements and/or can control such measurements. This synergistic and symbiotic feedback control loop between applications/simulations and measurements is a novel technical direction that can open new domains in the capabilities of simulations with high potential payoff, and create applications with new and enhanced capabilities. It has the potential to transform the way science and engineering are done, and induce a major beneficial impact in the way many functions in our society are conducted, such as manufacturing, commerce, transportation, hazard prediction/management, and medicine, to name a few.

Traditional application simulations are conducted with static data inputs. In the new dynamic, data driven application systems envisioned here, field collected data will be used in an "online" fashion to steer the simulations and vice versa the simulations could be used to control experiments or other field measurements. Thus the applications/simulations and the experiments (or field data) become a symbiotic feedback system rather than the usual static, disjoint and serialized approaches. The purpose of the workshop was to examine the technical challenges and research areas that need to be fostered to enable such capabilities. What are the requirements in the applications' level for enabling this kind of dynamic feedback and control loop? What are the requirements in the applications' algorithms for the algorithms to be amenable to perturbations by the dynamic data inputs? What are the challenges and technology needed in the computer systems areas to support such environments? The new set of applications will create a rich set of new challenges and new class of problems for the applications and systems' researchers to address.

Such challenges clearly present the need for a synergistic multidisciplinary research between applications and systems' and algorithms' areas. This research scope has the potential to help establish stronger and more systematic collaborations between the applications' researchers and the engineering, math and computer sciences researchers. How can such multidisciplinary research be programmatically fostered and supported in an effective way? How can this multidisciplinary research form a clear focus for many of the activities developed in existing individual programs supported in NSF? Past investments provide a basis to address the more challenging research required to enable the new paradigm fostered here. How can the research performed and the technologies developed under existing NSF efforts be poised to provide a relevant basis upon which the research for symbiotic measurement and simulation systems can springboard? How can the research focus for this new paradigm serve as a necessary adjunct of existing programs?

Many application areas can be envisioned benefiting or enabled from this new paradigm. Many are of interest to the research community supported by NSF. Representative examples were addressed in the workshop to illustrate the potential impact that this kind of research can have. The capabilities discussed here, are relevant not only to applications of interest to the NSF funded research community, but also to applications of interest to other agencies. Furthermore such new directions can provide a very positive impact with respect to the educational

---

<sup>1</sup> When **application** is referenced alone, we mean it to include all related application simulations as well as all other dynamic applications.

component, by providing the opportunities for students to work in some novel, exciting, and multidisciplinary projects.

The workshop took place at NSF and assembled about 80 scientists, representing relevant disciplines. The workshop addressed the problems, needs, possibilities and opportunities for such multidisciplinary research and education. These issues were discussed in the format of plenary sessions and breakout groups, along the areas of applications, algorithms, and systems' software technologies<sup>2</sup>. The present report summarizes the discussions that took place during the workshop, in order to make them available to the wider community, and to also serve as guidance for NSF's programmatic considerations.

#### The Workshop Co-Chairs

Prof. Craig Douglas, University of Kentucky and Yale University

Prof. Abhi Deshmukh, University of Massachusetts

#### Working Group Chairs

Prof. Michael Ball, University of Maryland

Dean/Prof. Richard Ewing, Texas A&M

Prof. Chris Johnson, University of Utah

Dr. Carl Kesselman, University of Southern California (ISI)

Dr. Craig Lee, The Aerospace Company

Prof. Warren Powell, Princeton University

#### The NSF Coordinating Committee

Dr. Frederica Darema - Chair

Dr. John Cherniavsky

Dr. Clifford Jacobs

Dr. Richard Isaacson

Dr. William Michener

Dr. Christopher Platt

Dr. Lawrence Seiford

Dr. Kamal Shukla

Dr. Roy White

---

<sup>2</sup> Slides of the invited presentations and Working Group summaries are posted on <http://www.cise.nsf.gov/eia/dddas/>

## Table of Contents

Preface.....	p ii
Table of Contents.....	p iv
Executive Summary.....	p 1
Main Body of Report	
Introduction.....	p 6
Applications	
General Properties of DDDAS	
Time Dependency and/Real Time Aspects	
Data Streams in Addition to Data Sets	
Combining Local and Global Knowledge	
Interactive Visualization and Steering	
Model Interactions and Software Agents	
Algorithms	
Data Driven Components	
Optimization and Inverse Problem	
Uncertainties in the Data	
Multiple Scales and Model Reduction	
Systems (Requirements of the Computational Systems)	
Physical Devices and a New Generation of Sensors	
Information Services	
Application, System, and Resource Management	
Programming Environments, Security and Fault Tolerance	
Economic Models of the Computational Infrastructure	
Why Now?	
Educational/Training Benefits	
Summary	
Appendix 1 / Workshop Agenda.....	p 22
Appendix 2 / Working Groups Charges.....	p 24
Appendix 3 / Application Examples.....	p 26

## Executive Summary

At a February 17, 2000 congressional briefing, meteorologists were asked why they missed predicting the track and magnitude of a major storm in January 24-25, 2000, that blanketed major cities from South Carolina to New England. One of the reasons cited by the scientists is that computer models (simulations) were not geared to incorporate changing conditions (like prevalent winds) as the many hours long computer simulations proceeded.

Even as the present report was in preparation, on May 7, 2000, the national park service started a controlled burn near Los Alamos National Laboratory. Within a day, the fire was labeled a wildfire. Once again, the existing methodologies were unable to simulate what the behavior of the fire based upon real-time changing conditions, and the emergency response agencies were thus unable to take appropriate and effective actions to limit the propagation of the fire.

These examples are neither isolated nor unique. Typically, applications and simulations we use today, only allow data inputs that are fixed when the application/simulation is launched. Traditionally, these processes are disjoint and serialized, not synchronized and co-operative. This lack of ability to dynamically inject data into simulations and other applications, as these applications execute, limits the analysis and predictive capabilities of these applications. Needs for such dynamic applications are already emerging in business, engineering, and scientific processes, analysis, and design. A number of examples of such applications are referenced in the main body of this report.

***The New Paradigm:*** *In the new dynamic data driven application systems framework envisioned here, the simulations and the experiments (or field data) become a symbiotic feedback control system.*

The primary objective of the workshop discussions was to identify research opportunities for the development of applications and system's software technology enabling this new generation of dynamic/adaptive applications. The novel capabilities sought, are simulation applications that can dynamically accept and respond to field data and measurements, and/or can control such measurements in a dynamic manner. This is a new dimension in the capabilities of applications.

The needs for enabling the new paradigm push for research leading to **leap ahead** technology capabilities. For example, to enable the kinds of application simulations discussed here, progress is needed in application methods and interfaces, and in algorithms that are tolerant to dynamically steering the simulation. Therefore research in the development of new such methods and algorithms for the specific application areas will be needed. Furthermore, the dynamic application requirements will dictate computing systems' support that includes systems' software technologies, such as active middleware services for real-time, dynamic reconfiguration capabilities, resource discovery, load balancing, security, fault tolerance, quality of service, and dynamic interfaces with field measurement systems. Currently the underlying systems' technology is not geared to support the dynamic requirements of these kinds of applications.

Therefore research is needed on: applications, for developing the dynamic, data driven application technologies, algorithms tolerant to perturbations of dynamic data injection and steering, and systems software for supporting the dynamic environments of concern here. In turn, research and development on these technologies forms the need for synergistic multidisciplinary research between applications areas with systems and algorithms research, and involving researchers in engineering, basic sciences, math, and computer sciences, in multidisciplinary teams as well as individual research efforts.

The workshop<sup>3</sup> included plenary presentations on application case examples where the new paradigm creates additional capabilities and benefits. The working groups were organized around the themes of: applications, algorithms, and (computer) systems. Charges<sup>4</sup> were provided to the participants to drive their discussions. Specifically the workshop addressed the issues and possible research directions for areas such as the following:

Data driven components, data assimilation<sup>5</sup> and feature extractions, model enhancement for local resolution, optimization and inverse problems, inverse problems for fine scale application models, computation/model and measurement system interaction, computation and computational infrastructure interaction, time dependency and real time aspects, data streams in addition to data sets, uncertainties in the data, multiple scales and model resolution, model interactions and software agents, interactive visualization and steering, combining local and global knowledge, exploiting new generations of sensors, information services, resource, and systems management under physical systems and real time systems constraints, application management and dynamic application component assembly, dynamic programming environments, security, fault tolerance, and economic models for the computational infrastructure.

A more detailed discussion of these research areas is provided in the main body of the report.

In the main body of this report, we also give specific examples of the kinds of applications that were discussed at the workshop. Many of the application examples presented here are of interest to the research community currently supported by NSF, and they are provided to elucidate the potential impact that this kind of initiative can have, rather than being an exhaustive or limiting list. In addition, applications that can benefit from the DDDAS paradigm are not only the ones addressed by the research community funded by NSF, but also those of other agencies (e.g., DARPA, DOE, and NASA) who were represented at the workshop.

Multidisciplinary research projects and multidisciplinary teams will be crucial for developing, in an effective manner, the necessary novel methods, frameworks, and tools, that are required to realize DDDAS. Furthermore, this kind of multidisciplinary research will have a very positive impact with respect to the educational component, by providing the opportunities for students to work in some novel, exciting, and multidisciplinary projects.

Why now is the time for developing such capabilities? DDDAS is a powerful and new paradigm, requiring advances on several technologies. However over the recent years there has been progress in a number of technology areas that makes the realization of DDDAS possible. These include advances in applications and algorithms for parallel and distributed platforms, computational steering and visualization, computing, networking, sensors and data collection, and systems software technologies. These are some of the recent advances that can help enabling the new paradigm of DDDAS. It is necessary however to endow these technologies with advanced and enhanced capabilities to develop the state of the art in DDDAS, which in turn will enable applications that are more powerful, effective, accurate and robust than what is currently available today. Productivity and services will be improved as a direct result of this new synergy and advances in technology.

---

<sup>3</sup> See the workshop agenda in Appendix 1.

<sup>4</sup> See the list of charges in Appendix 2.

<sup>5</sup> DDDAS is not tautonymous with data assimilation (DA). In DA data from multiple sources can be fused and used as inputs into an application. However DA is a static process, in that these sources of the data are fixed when the application is launched. While data assimilation can be exploited in DDDAS environments, the difference between DA and DDDAS is that in DDDAS the data injection in the application is dynamic (at runtime). In addition in the DDDAS paradigm, the applications will be empowered to control the measurements' processes

If this initiative is successful, a revolutionary change can be expected in how applications and simulations involving time dependent data are designed and what they can accomplish. We can expect advances perhaps equivalent to what happened to the manufacturing industry after computers were introduced in the 1950's. With the DDDAS capabilities many fields will be positively affected or revolutionized, including those in the basic sciences, biology, engineering, and the social sciences. An initiative in DDDAS will have relevance and will affect all areas of NSF.

The multidisciplinary DDDAS initiative will form a clear focus for many of the activities developed in existing individual programs. In particular, research performed and technologies developed under a number of existing initiatives provide the foundation upon which to build the DDDAS initiative. One workshop conclusion is an encouragement to NSF to announce a initiative on DDDAS that will support 15 to 30 multidisciplinary research projects in this topic for a duration 3 to 5 years each. The workshop participants also believe that this should be a sustained effort and concluded that such a call for proposals should be renewed at least two additional times in a span of 12 or 18 months apart. The possibility of some kind of joint announcement involving other agencies (e.g., with DARPA, NIST, or DOE) would create a significantly larger budget that would result in starting a significant number of projects and should be considered.

## Main Body of Report

### Introduction

Now assume that we are a few years into the future and that dynamically data driven application systems have become commonplace. A disaster like the two cited in the Executive Summary need not be uncontrollable. For instance<sup>6</sup>...

Near Los Angeles, an unknown fault moves violently causing an earthquake of 8.2 on the Richter scale. Above the fault a dam disintegrates and billions of liters of water pour down a canyon towards people and a petrochemical plant. Due to the earthquake, gas lines rupture, causing numerous fires including a forest above the dam. Numerous chemicals' storage tanks are ruptured and begin to leak toxic chemicals into both the air and the ground near underground aquifers that supply potable water to populous areas as well as individuals. Highways buckle and collapse at numerous points, hindering response team action. Nearby airports sustain major damage to their main runways. The main transportation systems are left in a state of chaos. Underground fiber connection severance, results in local communications disruptions.

Due to dynamic data driven models incorporated by the water and chemical companies however, damage and death were minimized. As soon as the dam began to fail, sensors in the canyon started feeding data via fast wireless networks into a spatially distributed network of supercomputers, the majority of which are located away from the disaster zone. A computer model predicted where the water from the dam will flow and the rate of the flow, continuously updated GIS data that were broadcast on the emergency broadcast system throughout the Los Angeles region.

Similarly, the petrochemical plant had previously installed sensors in the plant and the surrounding subsurface, to meet EPA monitoring standards. The sensors are capable of tracking the contaminants as they spread, which enables a continuous updating of the computer generated three-dimensional map of the toxins. Evacuation plans are optimized dynamically using streaming data of where the toxins are propagating. Subsequently the cost of containing and subsequent clean-up of the leaking toxins is considerably reduced by having continuously running predictions of where the toxins are migrating over many time scales. Regional weather models, interacting with global models, are used to predict where the airborne pollutants will travel.

Using telemetry from under roads and pattern recognition codes<sup>7</sup> for monitoring highway congestion, a cluster of PC's is able to help direct emergency vehicles to optimal routes for their destinations. Using a tracking system originally designed for school buses, emergency vehicles are continuously tracked and optimal routes are relayed directly to the vehicles, using advanced software running on a cluster of PC's. In addition, the bus onboard computer monitors local road conditions and obstacles and helps the driver to navigate through tight spots and obstacles. Emergency medical and disaster relief teams interact through wireless video/voice/sensor communications with regional medical centers and field hospitals to provide time critical medical attention.

The dynamically data driven application simulation workshop considered how a set of static applications could change into significantly more useful applications involving unpredictable

---

<sup>6</sup> ...The example given here is an amplified and "into-the-future" portrayal of an actual earthquake that occurred in LA in the 1920's

<sup>7</sup> The terms "code" and "program" are used interchangeably in this report. "Code" is the term used by applications' scientists, "program" is the preferred term among computer scientists.

dynamic changes. A number of technical areas have to be addressed in order to transform the applications from a “static data collection and assimilation phase plus computation” to an environment where field data is collected or mined dynamically, computation routinely expects the data to change dynamically, and steering applies to both the data and the computation.

Not only do algorithms and application methods have to be enhanced, but the system tools (middleware) have to be designed so that issues like application and algorithm adaptivity, rescaling, computer resource models, security, and fault tolerance, are routinely and dynamically available to the applications as needed and such capabilities can be designed into systems from the outset.

In addition student education and training curricula that embody DDDAS concepts and technologies will enable students to be trained in these novel technologies and tools, and prepare them for working in such multidisciplinary application environments. So the educational aspects of such an initiative are extremely valuable and exciting.

## Applications (General Characteristics, Examples and Related technologies)

An extensive but not exhaustive list of applications that can benefit from the new paradigm is given in Appendix 3. In the following sections, some examples of these applications will be cited to elucidate the points made.

### Characteristics of a Dynamic Data Driven Application System; General Properties

Pictorially the new paradigm is shown below. The intent is to show the new and tight (feedback and control) interaction between the ongoing computations and the measurements (or field-data collections processes):

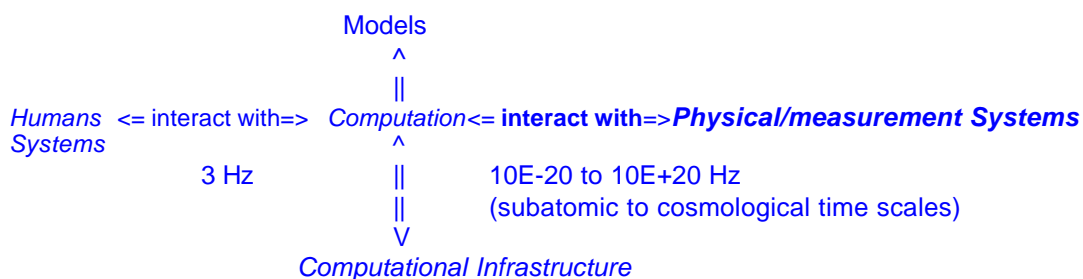


Figure 1: Diagram of DDDAS Interactions

Figure 1 identifies three primary modes of interaction for a DDDAS environment:

1. Human and Computation Interaction
2. Model or Computation Interaction with Physical System
3. Computation and Computational Infrastructure Interaction

While the DDDAS paradigm emphasizes the new technologies that need to be developed for modes 2 and 3, the program will be imbalanced if mode 1 is ignored. In fact by providing much more accurate information for the human in the loop, DDDAS will result in enhancing mode 1 also.

Physical systems (e.g., prosthetic legs, chemical plants, and active wings of an aircraft) operate at widely varying rates. Cosmologic and geologic rates are extremely slow relative to the timescale of subatomic events that happen very fast. Physical processes can also "produce" and



"consume" widely varying data volumes. In many cases, a computation may be able to interact directly with a physical system via some set of sensors and actuators (e.g., a prosthetic leg that can sense the terrain and apply the necessary forces to complete a walking motion). High-energy physics experiments provide another example.

For the purposes of explanation how this paradigm can affect other than engineering/scientific applications we provide here the following Supply Chain Management example from the ebusiness world (commercial/manufacturing and finance sectors): Businesses have increasingly become global but, at the same time, have become more specialized in order to concentrate on core competencies. As a result, the path from raw material to finished product delivered to a customer has become highly complex in terms of geographic scope, number of organizations involved and technological depth. Organizations have broadly adopted Enterprise Resource Planning (ERP) and Supply Chain Management (SCM) Systems in an attempt to better control and manage their dispersed operations. While these systems typically do well at collecting and integrating transaction data, and at providing support for planning decisions, they are not effective at supporting real-time decision making, particularly with respect to unplanned incidents. Fundamental research is needed into the modeling of the interactions among supply chain entities and the manner in which major events impact overall supply chain operations. Research is also needed into how to interface such models with the currently available data collection mechanisms. Under a DDDAS approach, a simulation would receive real-time data from ERP and SCM systems in order to maintain an accurate current picture of the supply chain. Decision makers would employ the simulation to project the impact of decision options. Of particular import, would be the analysis of ways to mitigate the impact of major disruptive events, such as emergency customer orders, plant breakdowns, missed shipments, etc. This environment has several distinguishing characteristics, including the wide range and magnitude of data sources, the fact that a variety of organizations own and/or generate the data and the dependence of system behavior on humans (decision makers and customers).

Humans will be increasingly interacting with physical systems through intervening computations. Humans can be considered as a multisensory system with a 3 Hz bandwidth, although some of its biological subsystems may have bandwidths which are considerably higher than this. Neurological activity, for example, can run at 2kHz. However, any human activity that involves cognition, will probably run (on the order of 3 Hz) much slower than almost any known computer or embedded system in use today. Other physical systems, however, may operate on a scale too broad, too fast, or too slow for direct interaction, as is the case in Cosmology. Computations must then be incorporated to approximate and interact with a mathematical model of the target system. In this case this necessitates dynamic data assimilation or dynamic data injection into the simulation to occur when the simulation finds that it needs more data from other sources. Similarly in the case where the simulations are used to control an instrument or other measurement device.

Computation is the instantiation of abstract notions that are modeled themselves on a *computational infrastructure*. We use the term computational infrastructure here in the most general sense: all machines and their connections, large and small. The range includes single machines to arbitrary collections of large, parallel machines and small, embedded systems as well as systems with high bandwidth and low bandwidth mobile dedicated communication.

Also in the most general sense, computations will also need to exploit dynamically the computational infrastructure on which they are running. A computation will also interact with a physical system via its computational infrastructure. The diagram above is a simple reference model pictorializing the environments considered here. In an actual instance, each of the elements could be multidimensional, e.g., there could be multiple physical systems and models interacting with multiple, distributed computations that interact among themselves and with multiple physical devices collecting and streaming data, and with multiple distributed humans. This representation of DDDAS problems guides us to emphasize and focus on several key characteristics that need to be addressed.

## **Time Dependency and/or Real Time Aspect**

The dynamic nature of DDDAS problems requires us to address the time dependency or real time nature of the applications. Certain applications (e.g., short lived physiological processes or sporadic astronomical phenomena) require real time response to observations from experiments or field data. The data driven aspect of these problems pertains to the closed loop between applications, algorithms and data. The incoming data stream can be used for dynamic decision making and/or for adapting the underlying models of the phenomenon.

Almost any dynamically data driven application simulation raises the issue of real time results. This is not always the case, however. Simulations can run in faster or slower than real time. This impacts the data rates required so that the new data must be introduced into the simulation in a timely and appropriate fashion.

In weather prediction, it is common to run simulations for up to five days as a batch process. The individual application simulation periods are a few wallclock hours, but do not update the early time steps with real data as it becomes available. Modifying weather prediction application programs to incorporate new dynamically injected data, is not a small change in the application program. It requires rather a fundamental change in the application design, the underlying solution algorithms, and the way people think about the accuracy of the predictions.

## **Data Streams in Addition to Data Sets**

The use of continuous data streams presents an additional challenge for data driven simulations since the results vary based on the sampling rate and the discretization scheme used. In other cases dynamic data assimilation or interpolation might be necessary to provide a feedback to experimental design/control. DDDAS algorithms also need to dynamically assimilate new data at mid-simulation as the data arrives, necessitating “warm restart” capabilities. Relevant semantics, ontologies and structure issues need to be addressed [McRae’s talk].

Data inputs to the dynamic data driven applications may be in the form of continuous data streams in addition to discrete data sets. Incorporating discrete data inputs during execution of an application itself presents several challenges, such as the ability to warm start the algorithm when new data arrives or to guide the search using new information. The use of continuous streaming data requires the algorithms to use appropriate data discretization methods to use the available information. The issues of optimally discretizing continuous data and providing feedback to the data generation process, either sensors or computational code, to change the sampling frequency are inherently interesting research issues. Moreover, data driven applications will not always receive data from known sources with well defined structure and semantics. The ability of handle different data structures and elicit appropriate semantic information is crucial to a robust DDDAS.

Consider the example of forest fire control [Coen’s talk], several low cost sensors could be dropped in the fire prone area to continuously monitor environment variables. The data gathered from these sensors will be incorporated in the simulation models of the affected region in order to accurately predict flare ups or unexpected changes in the fire frontier movement. The applications must incorporate the changes in the environment variables, without a cold restart of the simulation, in order to accurately predict behavior of the forest fire in time to allow corrective actions by the fire fighters. The accuracy of the forest fire behavior prediction and the response time available to the fire fighters is directly correlated to the ability of the dynamic data driven simulation code to incorporate incoming data at optimal sampling rates.

Another example where unstructured data may be streamed continuously in a computational code is from transportation modeling [Powell’s talk]. A computational model for dynamic vehicle

routing over a highway network can be interfaced with the routing information from individual vehicles. The decisions made by each driver on the route, start and end locations, and the driving conditions differ for each vehicle. In order for the traffic simulation model to incorporate all the information from each vehicle, it must be able to handle different pieces of information at different points in time.

Traffic light control is by itself an interesting DDDAS problem since there are two significant variants: is the plan to minimize or maximize the number of red lights encountered<sup>8</sup>? Many communities wrestle with this decision and how to optimize the timing of the lights continuously. Data is constantly generated from sensors under streets. The more sophisticated systems predict vehicle movement based on additional factors such as the weather. Until recently, a large set of mainframes operated traffic control systems. It is becoming much more common now to see a cluster of PC's running an entire system for cities of up to a few million.

### **Combining Local and Global Knowledge**

Combining local knowledge (observations) with global data for system level inferences: This relates to the earlier issue, where the autonomous subsystems possess local data and this information needs to be synthesized in order to obtain system level predictions. Some of these issues were discussed earlier on in this section. In addition there are issues of dealing with the various data models of these multiple data sources, and enabling application interfaces to these heterogeneous data models.

**Tightness of Feedback:** If both sensor and actuators exist, there can be a feedback control loop. The key question is how responsive must the computational architecture be to correctly interact with the physical system?

### **Model Interactions and Software Agents**

Modeling of transportation systems [Powell's talk] is once again an ideal example for modeling interaction between autonomous systems. Each driver makes a route selection based on personal travel plans. However, the interaction between different vehicles needs to be modeled in order to estimate congestion on the highway network. A dynamically changing simulation model of the transportation network needs to be able to accommodate a variety of vehicles and traveler profiles in order to accurately estimate the congestion and prescribe corrective actions.

Modeling interaction (data exchange) between autonomous (multi-agent) systems: Several applications can be modeled as compositions of autonomous systems that interact. In order to understand the behavior of the overall system the data exchange between the different elements needs to be specified.

Interactions between different subcomponents of a complex dynamic data driven system may not be defined a priori. Several applications can be modeled as compositions of autonomous systems that interact. This representation of a complex system may be dictated by the application environment, such as a transportation network where each vehicle represents an autonomous decision maker, or by the need to decompose the problem into computationally tractable reduced or decomposed problems.

In order to understand the behavior of the overall system, the data exchange between the different elements needs to be captured. Moreover, the subcomponents must be designed with the capability of interacting with a wide range of loosely coupled systems. The fundamental research problems at the heart of this issue are related to allowing codes to have flexibility in

---

<sup>8</sup> minimize (i.e., traffic control) or maximize (i.e., they would not be called stop lights if you could pass one without stopping first)

interactions with other elements of the system based on the state of the system, decision processes which allow each subsystem to determine appropriate interactions at each time step, and the ability to predict the overall system behavior based on the interaction models and the functions of each subsystem.

## **Interactive Visualization and Steering**

Visualizing complex high dimensional data in order to support human decision makers: Visualizing output of a complex model that responds to incoming data is critical when humans are involved in the loop.

While the need for the "human in the loop" may be diminished or eliminated from many simulations with advanced DDDAS techniques that we envision, there are many applications where human intervention may be necessary. In many cases, people are still needed to steer simulations because the results lead to immediate questions that can only be answered by a human expert, inspecting many possibilities. While techniques such as game theory can be applied to sort out good and bad branches to get optimal results, it is not practical if each branch of a decision tree takes several wall clock hours, weeks, or months to investigate. People can look at a simulation at one of the decision branches and an expert can steer the simulation in the direction of a "good" enough solution.

Three dimensional (in space) visualization is a common requirement for simulations. Many methods are in place now for viewing three dimensions of multiple scalar and vector variables on a two dimensional screen. Adding a fourth dimension (time) requires animation technologies to be commonplace in the future. Software will need to be developed and standardized in order to realize this task and make it easily approachable by newly trained developers and users. DDAS can build on existing NSF programs in this area and provide new areas of research in return.

## **Algorithms**

### **Data Driven Components**

The LA disaster case example in the Introduction is driven by continuously running, coupled geological, weather prediction, structural, and transportation simulations. Simultaneously, societal concerns, time critical dynamic responses, and data intensive simulations are run on a multiscale basis, and must interface and interact with earthquake models, regional weather models, and both surface and subsurface flow models.

We must identify important information from a discrete event or through data mining of sensor input or simulation output. This will involve some or all of the following techniques:

- Dynamic injection of data into the application
- Data assimilation and feature extraction
- Visualization with a human in the loop
- Algorithmic tolerance to perturbations by streamed data
- Sensitivity analysis

#### Dynamic injection of data:

Application interfaces need to be developed that allow ability of streaming of field collected data into the application at execution time. Employing techniques like data compression and striding through data might be needed depending on the application or the data collected. The ability of switching between such modes as the application executes might also be necessary, in cases

that would be dictated by the application (e.g. too many data – needed fast – compress; or too many data – need results fast – stride and sample over some of the data)

Dynamic Data assimilation and Feature Extraction occurs when there is dynamic or extra data available to a simulation. Weather prediction is a good example of these features. A simulation predicts the future, using a specific input data set when the simulation is launched. However field data can be generated or acquired continuously, even after the simulation starts. As real data becomes available, the simulation can incorporate it into the model and recompute the predictions. Ideally, a model would exist so that given a prediction, the simulation can be run backwards in time to see if the initial data is matched [Jones' talk].

Visualization with a human in the loop Many simulations require a visual approach with a human offering feedback to the simulations. Weather models and flow simulations frequently produce a computer animation showing important dynamic features. A human can steer the simulation to more fully investigate interesting features that develop. As part of a dynamic data driven simulation, the continuously arriving data will enhance, rather than eliminate, the human from the loop.

NOTE: Also we want to distinguish here the concepts of visualizing the new data and comparison of simulation output data from the concept of injecting data into the ongoing simulation.

Algorithmic tolerance and sensitivity analysis: Methods are needed to enable the application algorithms to be tolerant to the perturbations under streamed data. These items are discussed in more detail in subsequent sections, and include addressing multiple scales in models, algorithms, and data, with the realization that the scale of each can change over time. Enabling such capabilities will require:

- Model enhancements for local resolution
- Inverse problems for fine scale models
- Local gridding

Model enhancements for local resolution are common. In the LA disaster example, once the toxins have been located, more data can be collected in that specific geographical area so that a higher resolution picture can be constructed of the rate of accumulation of the toxins in that region. Inverse problems for fine scale models frequently have to be constructed and solved. This allows us to optimize parameters in models that are required before we can continue a simulation. Local or variable gridding allows for localized computing where interesting features are present. In the disaster example, we can use a coarse grid for most of the earthquake area, but we need very fine grids will be needed near the fault line. Similarly, we need a refined grid along the edge of the toxin flow region, but not away from there or in the center of the toxins.

On a higher level, requirements of models and data must be developed to produce the appropriate scale for interaction. Interface methods are needed to connect between these two aspects in order to get just the right scale and data, so that simulations are accurate enough without consuming too many computational resources.

Asynchronously collected data must be incorporated into dynamic data driven application simulations. In the LA disaster example, data deployment should be done periodically. Statistical errors in the data must be assessed and handled. Dealing with errors that are out of an acceptable range could be addressed with additional data collection or via tolerance built into the algorithms, or in some cases human intervention may be initiated.

For example, in the case of the forest fire, surveillance data can be collected only while the plane flying over the region [Coen's talk]. In particular in this example proper placement of sensors is paramount to useful data driven simulations. In the case of the disaster example, sensors can be dropped from an airplane flying above the region of interest. Small sensors

currently exist that propagate underground and are able to provide real-time measurements of temperature, wind, location, and the presence of certain chemicals. These data must be dynamically assimilated into a simulation. Algorithms tolerant to such perturbations of the dynamically injected data are necessary. For grid oriented applications, a moving, unstructured grid of data is continuously updated and affects which algorithms are appropriate, and methods are needed to select these appropriate algorithms dynamically at runtime. Some other methods to use would be interpolation methods to assimilate the data and filters to denoise it into a form amenable to the error bound restrictions of the simulations.

Remediation and response procedures may need to be incorporated into simulations [Ewing's talk]. The strategies must be based on the resources available and necessary constraints, such as the following:

- Multiscale model utilization
- Short time event updates of the objective functional
- Self adaptive, dynamic control
- Uncertainties which drive the dynamics
- Warm restarts of algorithms.

Each of these strategies may need to be employed while optimizing tradeoffs between time criticality, model fidelity, and resource allocation. Moreover, the algorithms must be robust and fault tolerant.

## **Fault Tolerance**

Fault tolerant algorithms become essential in the DDDAS application setting. Many DDDAS applications will run under dynamic conditions: the applications requirements are dynamic, changing in time depending on the dynamic data inputs, and also the underlying computational infrastructure on which these applications run will in general be dynamically changing. DDDAS applications are expected to run long time periods in varied environments. The application programs and algorithms will have to be able to handle seamlessly data streams, handling the rates at which they are produced, handling situations where they are produced at higher volumes than consumed by the application, and handling resource availability, like the situations where processors, memory, I/O, network connectivity and bandwidth, may disappear from the computational infrastructure (either for a short period of time or even permanently for the duration of the computation).

New systems need to be developed to allow continuing the execution of the application with a smaller number of computational resources. Today warm restarts provide a means for addressing this problem, this however is a limiting methodology and more dynamic and adaptive methods need to be developed and supported by the computational infrastructure, and not just for space-time grids.

An interesting research area will develop as security algorithms evolve that will permit programs to challenge data streams based on their content or their source. In the case of telemetry coming in from an oil field [Ewing's talk], there is a lot of room for deceptive data being delivered by a competitor who shares the oil field. Not only situations of terrorism [Powell's talk] need be considered as security threats, but over zealous competitors may in fact be a greater source of harm. Planning for these conditions is to some extent addressed in simulations today, but will be incorporated into DDDAS simulations in a more integrated way.

## **Optimization and Inverse Problems**

The evolving or adaptable nature of most DDDAS applications presents challenging opportunities for solving the associated optimization or inverse problems. Most simulations include some set of parameters that must be estimated in advance. However such optimal, or even near optimal selection is rarely possible. Furthermore, the optimal choice usually changes during the course of a long simulation. Inverse problems are designed to help in selecting appropriately such parameters. However, many inverse problems are unfortunately ill posed or extremely difficult to solve.

Optimization techniques for many simulations are not currently possible due to the increased size of the resulting problems. Better algorithms need to be designed and analyzed for very large scale problems that until recently have been too large to experiment on. With the current computing environments and platforms, such as for example the two NSF PACI leading edge sites and their partners, as well as future opportunities on which to leverage on, ample number of cycles should be available for such experimentation, at a level that has not been possible to date. In addition supercomputing management systems, like the Condor (system developed at the University of Wisconsin), make available the nearly infinite unused workstation cycles as well.

Consider trying to model the production processes in a chemical plant. Many parameters are necessary. Typically, a subset tries to maximize a particular set of products while staying within a set of constraints. Traditionally, many simulations are run using slightly different input parameters on static data sets. After a certain amount of computer time or wall clock time, the best set of parameters is used to continue simulations and to make decisions concerning the operation of the plant. A DDDAS version will determine the parameters using a data stream. Improvements in algorithms for inverse problems or for optimization algorithms that scale better than current ones are essential. Further, all of the parameters should be optimized, not just a very small subset as is typical today with data set oriented simulations.

This research needs to have a synergy among all of its practitioners, including applied mathematicians who are willing to work with simulation experts. This is an area where theoreticians can interact with computational scientists to address a number of challenging problems with respect to algorithms tolerant to dynamic data injection perturbations, making a significant impact on applications' capabilities.

### **Uncertainties in the Data**

Uncertainties in DDDAS applications emanate from several sources, namely uncertainty associated with the model, uncertainties in the input data (streamed), and the environment variables. Identifying the factors that have the greatest impact on the uncertainty output of the calculations, is essential in order to control the overall processes within specific limits. Computing all output distributions to provide error bounds is, for most realistic problems, a computationally prohibitive task. Hence, using prior observations to guide the output distribution estimations presents a possible approach to incorporating uncertainty in control decisions.

Incorporating these statistical errors (estimations or experimental data uncertainties) into computations, particularly for coupled nonlinear systems, is difficult. This is compounded by the fact that tolerance may also change adaptively during a simulation. Error ranges for uncertainty in the data must be created and analyzed during simulations.

Sensitivity analysis must be performed continuously during simulations with options in case of a statistical anomaly. Filters must be used, which are based on the sensitivity analysis in order to massage the data into an acceptable range. In many cases, the filters will need to be created as a result of applications or simulations moving from data sets to data streams. Data assimilation, Bayesian methods, non-linear multiresolution denoising, and Monte-Carlo methods are all candidates for sensitivity analysis and data filtering.



The common mathematical model in many DDDAS applications may be formulated as solving a time dependent, nonlinear problem of the form:  $F(x+Dx(t)) = 0$ , by iteratively choosing a new approximate solution  $x$  based on the time dependent perturbation  $Dx(t)$ .

In practice, the data streaming in may have errors and therefore may not be completely accurate or reliable (for example, in reservoir data sets, a 15% error in the data is common). As a result, perhaps one does not need to solve the nonlinear equation precisely at each step. This can expedite the execution

At each iterative step, the following three issues may need to be addressed. Incomplete solves of a sequence of related models must be understood. In addition the effects of perturbations, either in the data and/or the model, need to be resolved and kept within acceptable limits. Finally, nontraditional convergence issues have to be understood and resolved. Consequently, there will be a high premium on developing quick approximate direction choices, such as, lower rank updates and continuation methods, and understanding their behavior are important issues.

### **Multiple Scales and Model Reduction**

Multiresolution capabilities (scaling for multiple levels of resolution) are essential for DDDAS problems. Not only to negotiate scale between applications and data, but also to design efficient and adaptive solution methods. The ability to define different regions with differing granularities provides the decision makers with the ability to focus resources on critical areas (for example in the fire-fighting example, into regions where flare ups are highly probable).

The physical phenomena governing most of the applications discussed in this report may be extremely complex, with several parameters required to specify the governing equations. Incorporating automatic model reduction into solution procedures provides an additional means of increasing computational efficiency by lumping parameters, and simplifying basic principles. Developing high fidelity descriptions of the entire system may be computationally intractable. Several approximation and problem decomposition approaches may need to be developed and evaluated for complex dynamic data driven applications in order to select an appropriate method and model for a given application domain and parameter range.

Perhaps a straightforward approach would be a perturbative approximation of the existing applications and simulations approaches. However, for DDDAS, making these descriptions perform effectively in a symbiotic context is crucial. One goal of a DDDAS application is to have a number of descriptions available. Any one of these could be selected at a time in order to encapsulate the physical phenomena at any given instance. Having multiple descriptions available allows for testing to determine under what conditions a switch to another description is necessary or desirable.

One approach is to provide multiple resolution capabilities in the models, which allows both scaling (finer or coarser resolutions) of feature resolution in the same execution. Modelers can focus on areas where interesting or critical dynamics are observable, by using different scaling or granularity levels for different regions of the applications, as needed by the dynamically injected data. Multiresolution methodology provides means to identify scale -dependent features.

The ability to model systems with different resolutions can also be achieved by model reduction methods, such as lumped parameter systems. For example, in a simulation of an airplane, the model of airflow close to a wing is very complicated. However, the airflow sufficiently far away from the vehicle is a simple equation that represents constant airflow. Reducing the model away from the wing makes good sense and is a method that has been successfully applied.



Many problems are decomposed using similar or identical physical models in different parts of the domain and then tied together at the interfaces. Ocean modeling is frequently done by stacking several shallow water problems and allowing the shape of the layers to shift during the course of a multiyear simulation. Problem decomposition methods are frequently used to reduce the computational complexity. However, ensuring data integrity in the decomposed problems, especially when the algorithms are driven by constantly changing data, is a difficult problem which requires novel and accurate error tracking methods.

In the fire fighting example [Coen's talk], several physical phenomena need to be coupled in order to predict the behavior of forest fires accurately. For example, combustion models, heat transfer and fluid flow models, and structural models for the trees and the terrain need to be coupled in order to predict flare ups and the direction of the fire zone movement. The fidelity of models used in this example depends on the types of decisions being made. If fire fighters are using airborne chemical dispensers, then an approximate determination of the fire zone is sufficient. However, if several fire fighters are in the fire zone using ground based fire control methods then the prediction of flare ups needs to be accurate and is crucial in saving lives. Modeling the entire fire zone using a high fidelity model may not produce results in real time to take corrective actions. The ability to selectively model regions of the fire zone that have the highest probability of a flare up at high fidelity is highly desirable in such situations. These regions can change dynamically based on incoming data from sensors surrounding the fire zone.

## **Systems Infrastructure**

Computational systems for DDDAS require a fundamental advances in a number of areas. Aspects involved include physical devices, information services, resource, system, and application management, programming environments, security, fault tolerance, and economic models for the computational infrastructure.

### **Physical Devices and a New Generation of Sensors**

Physical devices will include not only processing and communication hardware, but also sensors and actuators. Sensors and actuators allow traditional computational devices to interact with physical systems. These devices are additional "resources" in the computational grid, and as such, resource discovery and allocation of sensors and actuators becomes an important issue.

In recent years miniaturization of almost all forms of electronics has led to a revolution in sensors. Global positioning systems, embedded into sensors, provide a new form of information generation that can be used in countless applications. We expect sensor technology to play a major role in measurements and field data collections for DDDAS environments.

When fighting a fire or trying to locate where a chemical contamination is moving and the toxicity propagation, inexpensive sensors that can broadcast a limited amount of information (temperature, location, the presence of a very limited number of chemicals, etc.) now exist that can be scattered across a region. When the sensor ceases broadcasting, this is an indication that either fire or chemicals may have destroyed it. When a collection of sensors can be used to form a clear pattern of certain environmental conditions. In the case of the specific example here, even more verifiable information can be added to the simulation of the disaster propagation, by including the fact the these sensors ceased indicating the propagation of destructive chemicals or fire.

Many sensors today are small and mass produced, easily specialized, and provide data cheaply. Small, portable sensors that interact with the GPS system provide a way of delivering data from most locations with great accuracy. Some sensors now can broadcast wirelessly for a short distance providing location, temperature, and a small amount of chemical data. These

sensors are ideal for providing data where people cannot or should go, such as into a wildfire or a major pollution area. Sensors are also becoming essential in medical procedures, such as brain surgery and microsurgery. Being able to place fast computers that provide visualization for doctors within their grasp is revolutionizing medical procedures. Further, data driven sensors for aiming devices for brain surgery are becoming more commonplace, and this research can be leveraged and applied to fields other than medicine. DDDAS will exploit these new advances in sensor technology, in a very effective way.

## **Information Services**

Added to the challenge of developing DDDAS is the realization that these more complex applications will have to function in a complex and perhaps evolutionary hardware and software environment. Present and future computing platforms, systems' software and applications' software will span a multi-part "globally distributed" computing environments environment {referred to as Computational Grids} that encompass the concepts of the meta-computing, heterogeneous hardware and software, networked and adaptive platforms, and will be manifest in a configuration ranging from assemblies of networked workstations, to networked supercomputing clusters. For example one of the novel and promising, as well as challenging, aspects of DDDAS, is employing an heterogeneous platform environments that include, but limited to, embedded sensors for data-collection, distributed high-performance simulations environments and special-purpose platforms for pre- and post processing of data, e.g. data assimilation and visualization.

For these kinds of platforms the underlying computing and communications resources available to the applications may vary even as the application executes. At the same time the dynamic data-driven applications will also have varying requirements as the computation proceeds, and therefore the resource requirements of the applications also vary. So with such variation in the underlying platforms and in the applications requirements themselves, as well as considerations of optimized performance and fault-tolerance, it may be required that the mapping of the applications on these platforms to change as the computation proceeds. Mapping the kinds of applications of concern here on these platforms requires dynamic and enhanced (dynamic/adaptive/active) systems services, which will need to be developed to allow DDDAS to effectively operate in the complex and heterogeneous computational environments that are emerging and will exist in the future. Fault tolerance and Quality of Service (QoS) will pose additional challenges for the applications software needed and/or the middle-ware services. The kinds of environments we consider will most likely include internet connected resources, and therefore addressing issues of scalability will become crucial challenges for the development and distributed execution environments envisioned here.

Computations must be able to be cognizant of and able to exploit the computational infrastructure on which they are running. Hence, information services must be available, such that an application can discover not only other compute resources, but also models for all manner of data and services that are available in the computational infrastructure. Such data models can include concepts such as uncertainty or any quality of the data.

Different application domains may need to catalog and identify alike resources in entirely different manners. Hence, different naming schemas or ontologies, information services, or information views may be necessary to properly support different application domains. These information services need to be distributed such that information locality is maintained in much the same way as data locality. This is necessary to ensure good performance and low latency access to fresh resource data or information. This also helps in achieving a high level of integration and responsiveness between computational and physical systems. The distributed nature of information services is also essential, and it will be necessary to ensure scalability of the computational infrastructure.

## **Application, System, and Resource Management**

An application may need to move code or data depending on runtime conditions. A typical tradeoff can occur if it is cheaper to process a large data set locally rather than paying the network overhead to move the data to a faster host. Remote visualization of large data sets is an example application for these tradeoffs. Making this decision involves being able to evaluate a complexity model for processing versus the data transfer time. Applications may need to reorganize data on the fly using network caches. Applications may also have to reconfigure to cope with resource failures or to make better use of available resources.

Just as the hardware needs to be monitored, so does the basic system software and middleware. Fault tolerant applications provide a challenge to middleware. Which systems should be used so that if a processor drops out of a computation, the computation does not hang and have to be run again. The more fault tolerant a system is, the higher the overhead usually is. There has to be a level of risk that an application or simulation takes that produces results quickly most of the time. However, once there is a fault, the level must be reducible (and later increasable) in order to get the right risk scale. New system management tools that allow discovery and advice on how often data is polled, or backed up to another system (which may be another processor or storage device), and how processors allocated or utilized will be imperative to DDDAS research.

Together with managing the underlying computing and communication platforms resources, sensors, actuators represent the additional resources that need to be managed in the computational infrastructure for dynamic data driven applications. In order to efficiently allocate these resources dynamically in the execution of a computational scenario the performance of these resources must be monitored. Sensor data can also be treated as performance monitoring data. Performance and sensor monitoring must be multiscale, and it must be capable of handling different scales and granularity of data available depending on the "proximity" to the sensor and information needs. The use of sensors and actuators can be planned and scheduled through resource brokers or via negotiations between applications. The resource management schemes need to incorporate concepts of advance (in time) and immediate (capacity) reservations (QoS).

Economic Models for the Computational Infrastructure: Ultimately, comprehensive economic models will be developed for use of the computational infrastructure [Frank's talk]. This would not only include simple accounting, but also the cost and contracting for advance and capacity reservations. Economic models must permit explicit modeling of resource contention between different requirements in dynamic data driven computations. Further, any economic model should be available through the information services.

## **Programming Environments, Security, and Fault Tolerance**

Programming environments must be able to support all of the capabilities discussed above to make them easy to use. This requires libraries with well defined APIs, programming tools, and middleware. Examples of middleware include model resolvers that can enable translations and reorganizations of data streamed between computations based on the output and input data models. Tools must dynamically compose other services and tools. Besides making code development easier, this enables the development of visual programming environments.

Security should be designed as part of any system at all levels. Encryption, authentication, and authorization must be provided where needed. Secure mechanisms are needed for application access of the required computing infrastructure, with dynamic group security domains to support collaborative environments. System status monitoring should be available to notify applications of resource failures. Exact notions of fault detection, fault containment, and fault recovery should be application specific.

## Why Now?

Many application simulations today work in the batch world: an event is simulated based on a static set of data. If newer data becomes available, the simulation is simply rerun. Very few applications use real time data streams even if the capability to do so is available. Great efforts have been devoted in fields like weather prediction to run simulations faster than real time based on static data sets. Ensembles are produced to get an average guess as to the weather based on a number of parallel runs with small variations in the parameters. This is highly inefficient and leads to multiple weather predictions that are mutually conflicting when major events are predicted (e.g., compare Accuweather, CNN, and the Weather Channel for examples of snow predictions in the northeast).

The fastest computers today, including terascale ones, provide a level of service that has been dreamed of for decades in numerous scientific fields such as weather, climate, and whole industrial plant simulation. Presently, each of these fields can produce data streams using sensors that have been developed over many years. We are now poised to do real time, data driven simulations with feedback, warm restarts, and continuous updates.

Clusters of inexpensive, fast PC's are providing cost effective and scalable computing platforms that may well be the future of supercomputers. Clusters can range from a few processors up to thousands depending on the budget and floor space allowances. Once again, with enough processors, PC clusters can compete with the traditional supercomputers. However, issues like how to update a few thousand PC's at once need to be resolved before clusters will be truly competitive for DDDAS applications. These clusters can also be used for visualization and running data collectors. This is particularly important in medical applications [Johnson's talk]. New algorithms are being developed (or have been) using fast, new networks and the fact that there are cycles available on the nation's supercomputers that enable new ways of attacking old problems

A numbers of NSF programs today support batch style application simulations). DDDAS offers a plan for moving many of these simulations areas into the future where continuously fed data streams are the normal input instead of static data sets. Further, it moves the older style of data set analysis ahead by motivating people to keep a much larger data set with information from a wider set of times in libraries for debugging codes and developing new algorithms. This is similar to what has happened with single processor algorithms for fields where parallel computer algorithms are now standard: the older, easier to use serial computers have new algorithms that would not have been developed without the common usage of parallel computers.

There has been progress in a number of technology areas. These include advances in computing, networking, sensors and data collection, software, algorithms, and application technologies. Combining all of these technologies will lead to a higher level of application simulations that are both more accurate and able to provide analysis and prediction better than what is currently available in most fields today. Ultimately the effects of such applications can have impact on productivity that will be increased as a direct result of this new synergy of technology. DDDAS paradigm will reduce the time needed to adapt to new conditions and to decide how to allocate resources to respond to the unexpected, data dependent changes in simulations. This is particularly important in the following areas:

- Experiments on short lived processes (e.g., high energy physics and physiology)
- Capture of sporadic events (astronomy)
- Active control of environmental or safety controls in structures during an event (e.g., earthquakes or hurricanes)
- Disturbance in a chemical plant
- Early warning systems (e.g., weather, seismic, fire, pollution, tornado)

- Financial Systems
- Business, Enterprise and Manufacturing operations
- Medical applications

## **Educational/Training Benefits**

A research initiative in DDDAS will have tremendous benefit in the education and training of students, both at the graduate and the undergraduate level. The exciting new research areas, enhanced and even novel applications, novel algorithms, and new capabilities in systems software technologies, will provide fertile work grounds for students in the many applications disciplines, students in applied and theoretical mathematics for research to address the challenges of new algorithms and new methods to deal with the multiscale models and data dynamic data uncertainties. The work needed spans from the theoretical underpinnings for such systems to applied research on algorithms development and implementations. In terms of the computer science advances needed for developing the kinds of systems software capabilities that have been discussed here, a research initiative in this area will provide a wealth of exciting research projects opportunities for students to be involved and acquire the background on, a critical and valuable expertise in working on state-of-the-art technologies.

## **Summary**

The workshop addressed the motivations, challenges, and opportunities in supporting research that pushes towards new frontiers and creates a new paradigm for application and simulation systems. In this case, the simulations' input can be altered dynamically by real time field data and have such input dynamically steering the simulations. Additionally, the new paradigm seeks to establish capabilities where the simulations can be used to steer the experiments (measurements) or the field data collection or mining process. Such a synergistic feedback control loop between simulations and measurements is a novel technical direction with high potential pay off in terms of creating applications with new and greatly enhanced capabilities.

The DDDAS initiative will define new classes of simulation applications that are envisioned to have greatly enhanced capabilities than the present ones. From a research perspective, the initiative will define a new set of problems to be addressed and will create a strong feedback loop between the applications' research and the engineering and computer sciences research needed to support these enhanced capabilities. The DDDAS initiative will provide a fruitful ground for scientists to ask new questions that have not been addressed or even asked before. In fact, this initiative will help establish stronger relations between the applications' researchers and the engineering and computer sciences researchers.

The dynamic nature of DDDAS problems requires us to address the time dependency or real time nature of the applications. Certain applications, such as short lived physiological processes or sporadic astronomical phenomenon, require real time response to observations from experiments or field data. The data driven aspect of these problems pertains to the closed loop between applications, algorithms and data. The incoming data stream can be used for dynamic decision making and for adapting the underlying models of the phenomenon.

This initiative will provide an avenue for research that is not addressed by the current ITR initiative , but can easily be used to augment and incorporate possible ITR advances. There was a preponderant view of workshop participants, that DDDAS is sufficiently distinct in its vision from the broad goals of ITR to warrant a separate, more focused initiative.

## **Appendix 1 / Agenda for the Workshop on Dynamic Data Driven Application Systems**

National Science Foundation, Arlington, VA

March 8-10, 2000

The workshop will take place at NSF. All plenary presentations will be in Room 110.

Specific room assignments for Working Group meetings will be provided at the meeting

***Wednesday, March 8, 2000***

8:00a.m.-8:30a.m.	Registration and Coffee/Donuts
8:15a.m.-8:45a.m.	Welcoming Remarks by the Organizers, Co-Chairs and NSF Officials: Dr. Bordogna, NSF Deputy Director; Dr. Bajcsy, CISE AD; Dr. Eisenstein, MPS AD; Dr. Leinen, GEO AD; Dr. Wong, ENG AD
8:45a.m.-12:15p.m.	Plenary Presentations
8:45a.m.- 9:15a.m.	Greg McRae, Professor, MIT: New Directions on Model-Based Data Assimilation
9:15a.m.- 9:45a.m.	Janice Coen, Scientist, NCAR: Coupled atmosphere- wildfire modeling
9:45a.m.- 10:00a.m.	Break
10:00a.m.- 10:30a.m.	Howard Frank, Dean, Business School, UMD: Data/Analysis Challenges in the Electronic Commerce Environment
10:30a.m.- 11:00a.m.	Klaus Schulten, Professor, UIUC, Beckman Institute: Steered computing - A powerful new tool for molecular biology
11:00a.m.- 11:30a.m.	Dick Ewing, Professor, Texas A&M University: Interactive Control of Large-Scale Simulations
11:30a.m.- 12:00p.m.	Chris Johnson, Professor, University of Utah: Interactive Simulation and Visualization in Medicine: Applications to Cardiology, Neuroscience, and Medical Imaging
12:00p.m.-12:15p.m.	Charges to the Working Groups
12:00p.m.-1:15p.m.	Lunch
1:15p.m.-5:30p.m.	Break-out Groups (G1: Applications; G2: Algorithms; G3: Computing Systems)

**Thursday,**

**March**

**9,**

**2000**

7:45a.m.-8:30a.m.	Coffee/Donuts
8:30a.m.-10:00a.m.	Break-out Groups (G1: Applications; G2: Algorithms; G3: Computing Systems)
10:00a.m.-10:30a.m.	Break
10:30a.m.-12:00p.m.	Break-out Groups (G1: Applications; G2: Algorithms; G3: Computing Systems)
12:00p.m.-1:15p.m.	Lunch
1:15p.m.-3:00p.m.	Plenary / Interim Presentations of Break-out Groups' discussions
3:15p.m.-3:30p.m.	Break
3:30p.m.-5:30p.m.	Break-out Groups (G1: Applications; G2: Algorithms; G3: Computing Systems)

**Friday,**

**March**

**10,**

**2000**

7:45a.m.-8:30a.m.	Coffee/Donuts
8:30a.m.-9:00a.m.	Anita Jones, Professor, UVA: Injecting Simulation into Real Life Processes (Plenary Presentation)
9:00a.m.-10:00a.m.	Break-out Groups (G1: Applications; G2: Algorithms; G3: Computing Systems)
10:00a.m.-10:30a.m.	Break
10:30a.m.-12:00p.m.	Break-out Groups (G1: Applications; G2: Algorithms; G3: Computing Systems)
12:00p.m.-1:15p.m.	Lunch
1:15p.m.-2:15p.m.	Plenary / Final Presentations of Break-out Groups' discussions
2:15p.m.-3:30p.m.	Discussions on Presentations and process for completing the report
3:30p.m.	Meeting Adjourns

## **Appendix 2 / Working Group Charges**

### **Dynamic Data Driven Application Systems' Workshop Report**

- We have to write a report that summarizes the discussions of the working groups and plenary talks.
- The report should provide a compelling argument to senior management that research in dynamic data driven application systems should be supported now.
- The report will be online in the next 10 days. Suggestions for improvements will be welcome. Contributions will be even more welcome.

### **Issues for ALL Workgroups to Address**

- Why is now the right time to do this type of research?
- What ongoing research will provide a foundation for new research?
- What programs are necessary to enable the multidisciplinary type of research of this program?
- What initiatives exist that are related?
- What programming aspects should be emphasized?
- What exciting new opportunities will be created?
  - For postdocs?
  - For graduate education?
- How will it help industry?
- Technology transfer?

### **Working Group on Applications**

- Provide application examples that will benefit from
  - a new paradigm.
  - existing and new potential applications.
- Describe challenges in developing applications.
- What will be the composition of these applications?
- Describe data management and interfaces to experiments or field data.
- What are the computation, I/O, and memory requirements?
- What system support is needed in a dynamic application environment?
- What system resources must be allocated in order to assure a high quality of service?

### **Working Group on Application Algorithms**

- Describe the challenges in application algorithms to enable dynamic data driven application systems.
- What algorithmic enhancements are needed to allow perturbations in the data input?
- How do you factor statistical errors into algorithms?
- How do you change algorithms as data changes during a simulation?
- What knowledge based systems, interfaces, or application assists can be created to help with changing algorithms dynamically?
- What systems support is necessary?

### **Working Group on Computing Systems Support**

- What are the software challenges for programming environments for the development and runtime support when the underlying resources and application requirements can change suddenly?
- What capabilities are needed by the underlying systems that the applications will execute so that quality of service is high?
- What are the issues with respect to data management, models, and structures?



- What are the issues with respect to interfaces between simulations and measurements (and data issues)?
- What additional capabilities are needed for application support and systems management services?

## Appendix 3 / Application Examples

This section is under construction and will be updated shortly.

*Adaptive structures* (car suspensions, earthquake proof buildings, space structures)

### *Operational Systems*

Transportation and logistics, reconfigurable manufacturing, supply chains, autonomous (embedded?) systems, disruptions and disturbances in manufacturing processes.

A case example is provided by considering a typical chemical plant, where the fundamental chemical processes are governed by the underlying chemistry of the ingredients. However, in addition to the chemical rate equations, the reactor design and the operating conditions also affect the functioning and the yield rates of the process. In order to efficiently control a chemical reactor, we need to solve the coupled chemical kinetics, fluid flow/heat transfer and economic equations in the presence of uncertainty (discussed earlier). The use of incoming experimental data to guide the controller in chemical plant represents a perfect DDDAS application domain.

Air Traffic Management: The very significant growth in air travel within the US, coupled with the much slower growth in the implementation of new air traffic management and control systems, has led to substantial increases in air travel delays. Major delays on any given day are typically associated with particular events that have altered normal airspace capacity or demand patterns. Within the US, most delays are associated with disruptive weather events. A traffic flow management system must have the capability to predict capacity-demand imbalances and to support the generation of actions that mitigate the impact of these predicted imbalances. Examples of such actions include delaying flights on the ground, rerouting flights, canceling flights and restricting the flow in certain portions of the airspace. The DDDAS paradigm provides an ideal approach for creating a next generation air traffic management system that gives much more reliable predictions and also gives high quality support for traffic control and planning. Under a DDDAS approach a stochastic simulation of the National Airspace System would operate continuously and would be fed by a real-time stream of traffic status updates. The simulation would require models of both the en-route airspace, the airports, weather and airline responses to unfavorable events. This overall system would provide airspace status information and predictions to both the Federal Aviation Administration (FAA) traffic flow managers and to the airline operational control centers. Both of these parties would have access to models that estimate the impact of traffic flow manager control actions and airline flight-plan modification decisions. Such a system must react to data from sources that are both geographically and organizationally distributed (the airlines and the FAA). Furthermore, it must support similarly distributed decision making. Success in this problem area would have a significant positive impact on nearly all Americans.

Supply Chain Management: Businesses have increasingly become global but, at the same time, have become more specialized in order to concentrate on core competencies. As a result, the path from raw material to finished product delivered to a customer has become highly complex in terms of geographic scope, number of organizations involved and technological depth. Organizations have broadly adopted Enterprise Resource Planning (ERP) and Supply Chain Management (SCM) Systems in an attempt to better control and manage their dispersed operations. While these systems typically do well at collecting and integrating transaction data, and at providing support for planning decisions, they are not effective at supporting real-time decision making, particularly with respect to unplanned incidents. Fundamental research is needed into the modeling of the interactions among supply chain entities and the manner in which major events impact overall supply chain operations. Research is also needed into how to interface such models with the currently available data collection mechanisms. Under a DDDAS

approach, a simulation would receive real-time data from ERP and SCM systems in order to maintain an accurate current picture of the supply chain. Decision makers would employ the simulation to project the impact of decision options. Of particular import, would be the analysis of ways to mitigate the impact of major disruptive events, such as emergency customer orders, plant breakdowns, missed shipments, etc. This environment has several distinguishing characteristics, including the wide range and magnitude of data sources, the fact that a variety of organizations own and/or generate the data and the dependence of system behavior on humans (decision makers and customers).

Oil Exploration: Sophisticated oil exploration methods involve both measurements (seismic, sonic, and radiation propagation) in the subterranean regions of interest and extensive modeling and simulations of the measurement processes. Correlation of measurements with simulation analysis is used to pin point the regions of interest. Ability to inject dynamically measurement data to ongoing simulations can result in refining the ability of the simulation to pinpoint the thin petroliferous layers. Reversely when the simulation finds evidence of a potential petroliferous layer, the simulation can seek and use additional datapoints from data already collected or can control the measuring processes to collect additional field data as needed, thus making more targeted and efficient the measuring processes. This control loop between simulations and measurements will result in better analysis and prediction processes, and in reducing the cost of exploration, by expediting the measurements processes.

#### Natural Systems

Forest fires propagation and containment, weather prediction (extreme geospace conditions - space weather); hurricane (or other disaster) evacuation. Such examples have been discussed in the main body of the report and were the subject of some of the presentations at the workshop.

Radio astronomy needs a close connection between 90+ radio-telescopes to conduct a full sky survey. Data collected at one site needs to be analyzed (fed to a simulation?) in order to guide other telescopes to search in the appropriate orientation; Gravity wave detector and CERN particle detector (Eisenstein examples).

#### Social Systems

Policing and drug prevention measures, traffic control, terrorist attacks... (expand on the traffic control and terrorist attack examples).. monitoring of epidemic spread

Transportation systems and traffic control: Managing flow of traffic on a network, either on land or air, presents challenges similar to most social systems. Each individual entity (vehicle) decides on routes based on its origin and destination points. From a system level perspective, the excessive flow of vehicles on a specific segment represents a bottleneck situation. In order to model this application, the individual vehicle behavior needs to be considered along with the environmental conditions. Such a model could also be used to give specific early warning instructions to the vehicles in order to avoid congestion.

#### Human/Mechanical Systems

##### *Medical Device Design: Closed loop engineering design*

Sudden cardiac death kills approximately 250,000 people in the United States each year. The danger strikes when the heart alters its normal, steady beat and slips into a condition called fibrillation. Fibrillation describes a state in which the electrical activity throughout the heart is scrambled, resulting in the inefficient pumping of blood. Unless the fibrillation is reversed using an applied electrical shock, the condition leads to death. This shock is

administered from a device known as a defibrillator, typically administered by a highly trained health care professional, such as a paramedic or an emergency room doctor. These external defibrillators have been in use for some time, yet researchers have recently designed "implantable" defibrillation units that automatically detect and regulate an arrhythmic heart beat. Implanted within the patient's chest with electrodes placed near the heart, these devices first detect in its earliest manifestations the abnormal electrical activity that is the hallmark of fibrillation and then apply defibrillation pulses. As the heart begins to beat irregularly, the defibrillator applies a small jolt of electricity that returns the heart to its normal beat, typically before the patient even loses consciousness.

However, the design of such a device is very difficult. The device should deliver the proper electric shock to the heart, yet should not damage other internal organs. To complicate matters, many overweight individuals have an insulating layer of fat over their heart, rendering much of the electrical shock useless. There are many questions to be answered for the design of such a device: Where should the electrodes be placed? How many should there be? If I design one for Joe, will it also work for Mary? To help design such devices, researchers have used computer models of the thorax to simulate various configurations of electrodes and stimulation pulses and visualize the results using the SCIRun problem solving environment (see [www.sci.utah.edu](http://www.sci.utah.edu) <<http://www.sci.utah.edu>> for details). With SCIRun, engineers are able to imagine improvements to a device, and place them directly into the computer model. They will be able to test the new devices using the computer prior to animal and/or human trials, thus increasing the safety of the device. The use of interactive scientific visualization and simulation has been pervasive throughout this project and is opening up new ways of perceiving and investigating the complexities of physiologic systems. In addition, engineers can monitor patients (perspective candidates of such devices) and dynamically inject patient monitoring data into the simulation to further optimize and customize the design. When implanted into the patient the device uses monitoring data and to calculate and adjust the voltage, the frequency and the shape of the electric shock, depending on the instantaneous response of the patient.

#### *Artificial limbs*

Robot assisted surgery, medical imaging. A haptic device translates the surgeon's actions to motions inside a remote patient's body or in a simulated surgery. This capability can be potentially used to train surgeons in complex procedures, remotely perform surgical procedures and be used in microsurgery. This application requires integration of data from the sensors on the robotic device with the physiological model of the patient's body in real time. The ability to visualize the effects of the surgeon's actions in a simulated environment is extremely important in order to reduce in-process errors.

Link expensive data acquisition devices (e.g., high resolution microscopes); compare real data with simulation data and database information