Some initial thoughts on the proposal:

Since the proposed work will span a number of disciplines, it seems that a modular design of the various software and programs may be desirable. This could also be stated as one of the worthy purposes of the proposed research. By modular design, I am refering to creating a repository of reusable modules that are potentially useful to all those concerned with this project. The management of these modules may be best accomplished by relying on an object-oriented formulation of the problem. This brings out a number of interesting (?) issues such as OO.Databases, OO.PhysicalModeling, OO.FEM, OO.Parallel/Distributed.Computing.

Opting for an object-oriented formulation will require a closer interaction among the participating researchers.

Some issues to consider:

- Modules for system identification and control have to share the data with the modules for the prediction (FiniteElement, FiniteDifference, ...)
- Object Oriented description of the problem seems very natural - isolate "objects" of interest and track them with higher resolution. As I mentioned before, domain decomposition could be useful here.

- Parallelism of architectures and databases has to be focused on making the software (Finite elements, finite difference, system identification, etc...) more efficient.
- Adopt a description of the physical problem that is most efficient (natural ?) for parallel/distributed implementation.
  - Domain decomposition
  - Multigrid
  - Adaptive multigrid (for zooming in on some area of interest)
  - Object-Oriented formulation of the problem (Domain decomposition and Multigrid can play a central role here)
- Existing software to build upon:
  - PVM - Distributed computing
  - Khoros - Distributed computing, visual programming, signal/image processing, GIS
  - HDF, VHFD, NetCDF, - Database management systems
  - Grass, Arc-Info - GIS
  - Distributed and OODatabase System (any one in particular ?)
- Software to be developed as byproduct of this research (?)
  - Parallel and OO-GIS (maybe enhance ArcInfo and/or Grass)

– OOFinite Element / OO-Physical Modeler (based on Domain Decomposition ?)

We have strategy of using AVS as system integration software in many of application projects here at NPAC. Geoffrey recently suggested that we consider AVS as a porting tool for developing model systems, linking code modules that now exist in sequential/ parallel versions, different languages, best suited for different architectures.

This seems to be a practical, intermediate step to developling systems with existing software, useful as step to parallel and OO modules in a future improved system.

We have large nCube2 with parallel Oracle 7.0 software, and number of active high performance I/O parallel database research projects.

We have experience with GRASS software and plan to build parallel modules according to application requirements. Goal to maintain GRASS user interface, and manage heterogeneous computing environment to combine SIMD, MIMD, sequential machines.