3. Research Program Accomplishments

3.1 Program Overview

The central goal of the Center is detailed, whole-system simulation of solid propellant rockets under both normal and abnormal operating conditions. Full simulations of such complexity require a sequence of incremental developments—in engineering science, computer science, and systems integration—over an extended period of time. From the outset, however, our emphasis has been on *system integration* rather than separate threads of development that eventually come together at some point in the future. Rapid exploration of critical system integration issues entail the use of simplified—but fully integrated—models and interfaces initially, followed by successively refined models and interfaces as experience is gained.

Simulation Roadmap

The CSAR Simulation Roadmap (Fig. 3.1.1) depicts the evolution of increasingly sophisticated computational models for the primary rocket components and their interactions. The Project Timeline (Fig. 3.1.2) that accompanies the Roadmap indicates the time sequences required for the execution of the technical program. We have been remarkably successful in completing the tasks outlined for Years 1 and 2. Completed tasks are noted on the Timeline.

Our initial implementation of an integrated simulation code (GEN1), fully operational in Years 2-3, provides a simplified characterization of various burn scenarios to the possible onset of component failures. The GEN1 code employs macroscopic models for the separate components to enable a strong, early focus on the definition and resolution of system integration issues. Refined, multiscale component models and advanced system integration concepts, based on lessons learned from GEN1, constitute the key features in the second generation (GEN2) code—targeted for Years 4-5. The refined models also reflect the synthesis of fundamental, subscale studies (bottom right side of Fig. 3.1.1), which are critical for detailed simulations of accident scenarios and for reliable simulation of multiscale phenomena such as combustion and turbulence. The code numbers in the diagram indicates dependence of the refined and accident models on the subscale simulations.

The Roadmap also indicates the close coupling among the components; physical quantities such as temperature (T), mass flow ($\dot{m}$), pressure (p), heat flux (q), concentrations (c_i), and geometry must be exchanged between the SRB component models. The computer science integration efforts define the framework for these interconnections and, consequently, their eventual impact on overall code performance. In the right-center box on the diagram, computer science research and development activities are shown that support the SRB simulation through the implementation and optimization of the component models and subscale simulations, the integration of component models and the computational infrastructure required to do large scale parallel computation.

Finally, the central placement of validation efforts in the diagram highlights the priority assigned to this activity. Each subscale, component, and integrated simulation must be validated against existing analytical, numerical, and experimental data available in the open literature or obtained from our sister Center for Novel Energetic Materials (CNEM).

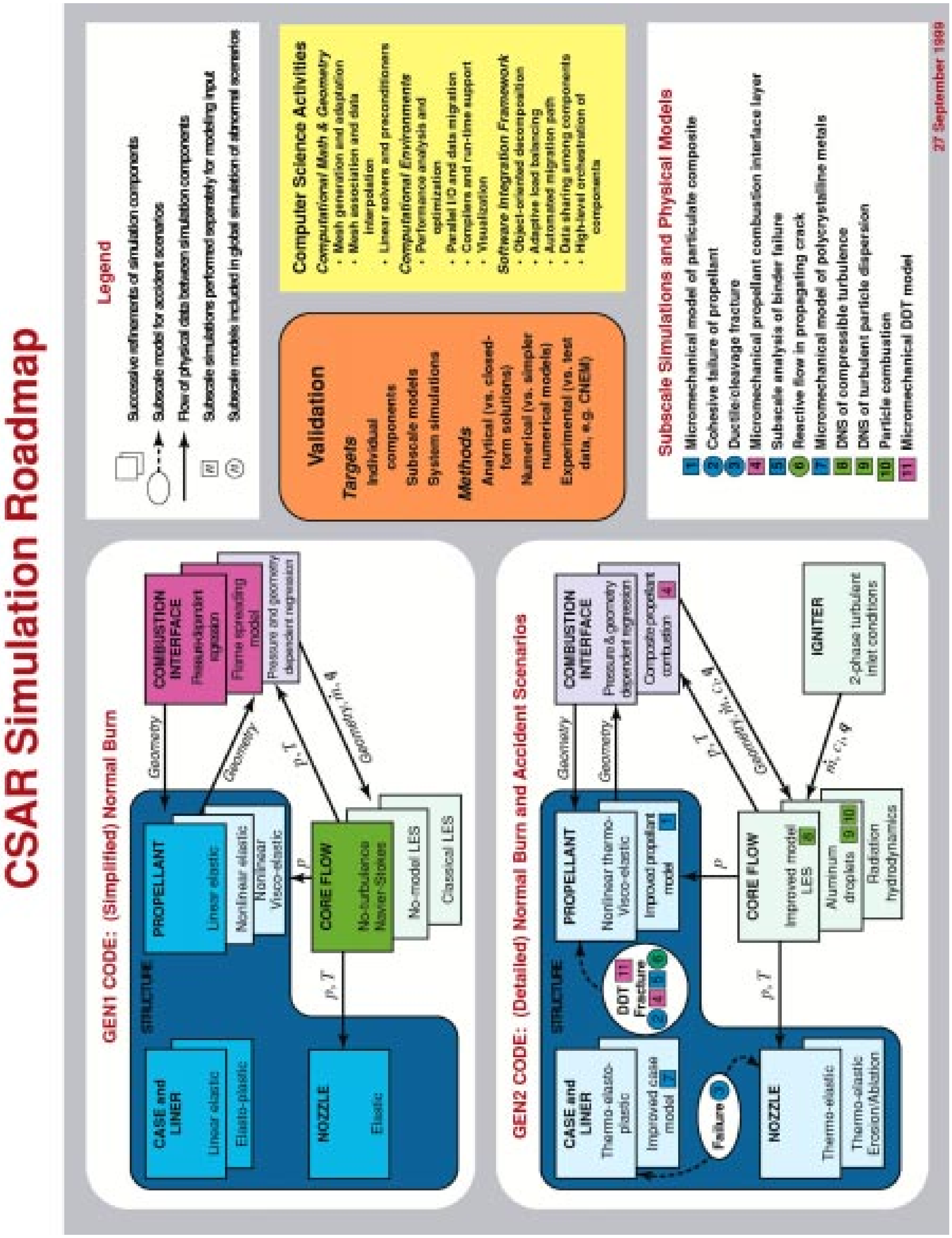

Fig. 3.1.1. CSAR Roadmap showing completed tasks (dark boxes in GEN1) and planned activities for Y3-10.

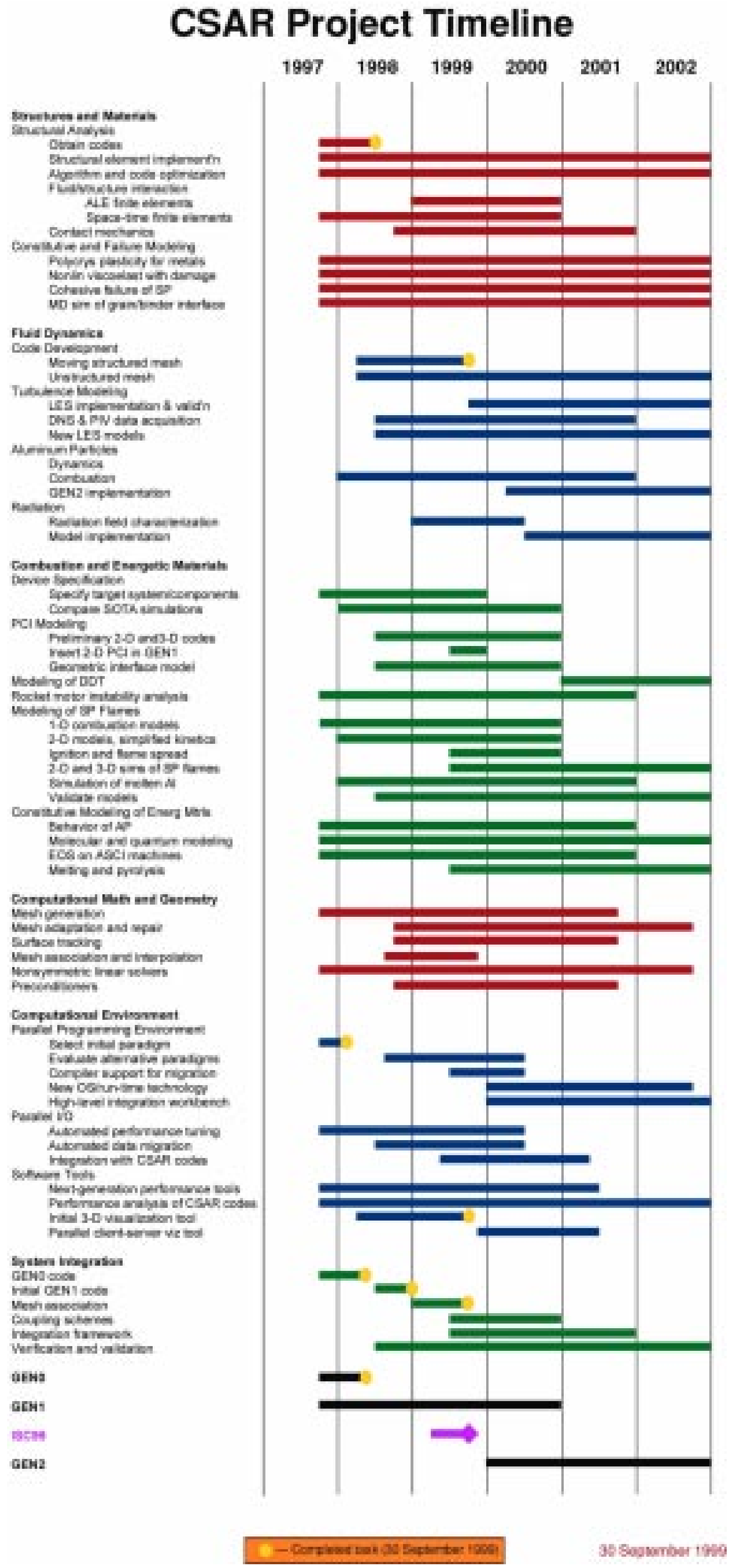

Fig. 3.1.2. CSAR Project Timeline shows completed tasks and future activities.

System integration involves two major tasks to ensure the physical, mathematical, geometric, numerical, and software compatibility of the component models and the codes implementing them. The first major task is providing information transfer across component boundaries. Boundary conditions for the component models must be compatible mathematically (e.g., an outflow from one component becomes an inflow for a neighboring component). The discretizations of neighboring components must fit together geometrically. Different spatial resolutions and discretization methodologies must be reconciled via interpolation where necessary.

The other major task is temporal coupling of the components so that the whole system is evolved in a self-consistent manner. Different components may have very different time step sizes due to the choice(s) of algorithm(s) (e.g., explicit vs. implicit methods), spatial resolution, and/or the physics of the subproblem that the module solves. The computational cost of forcing each module to take a time step determined by the module requiring the shortest step is often prohibitive. We continue to investigate multiple strategies for coupling modules requiring different time step sizes while maintaining the accuracy of the overall simulation.

Our approach to system integration has been to develop a single executable code containing modules for the various components and the interface code for tying them together. We are following an object-oriented design methodology that hides the data structures and other internal details of the individual component codes. This simplifies development and maintenance of the interface code and the component codes, and also makes it easier to swap different versions of the same component—a critical capability for determining the most efficient algorithms and implementations.

ISC99 Components

The ASCI Alliance Strategy Team (AST) and the DOE-DP Technical Support Teams (TST) identified a need to establish early and periodic prototypes of the integrated simulation capabilities (ISC) of each of the five university-based ASAP Centers. The goal for "ISC99" was a fully 3-D, coupled, integrated solid/fluid code, employing MPI parallelism on the ASCI machines. This goal was achieved in September 1999. The simulation included the following components and is described in greater detail in the remainder of this section of the CSAR Annual Report (see especially Section 3.6, System Integration).

Structures and Materials:

- Moving interface—regression due to burning of propellant
- Large model of the rocket with star grain in head end
- Implemented in F90; parallelism on element level; MPI for message passing
- Unstructured brick finite elements; multi-block mesh

Fluid Dynamics:

- Fully 3-D, including star grain and flow in head-end slots

- Moving boundaries with combustion-induced regression and structural deformation
- Automatic remeshing, including solution-based grid adaptivity
- Fully compressible core and nozzle simulation (no plume)
- Laminar Navier-Stokes equations (no turbulence modeling)
- Explicit multistage Runge-Kutta time stepping
- Implemented in F90 with MPI for interprocessor communication

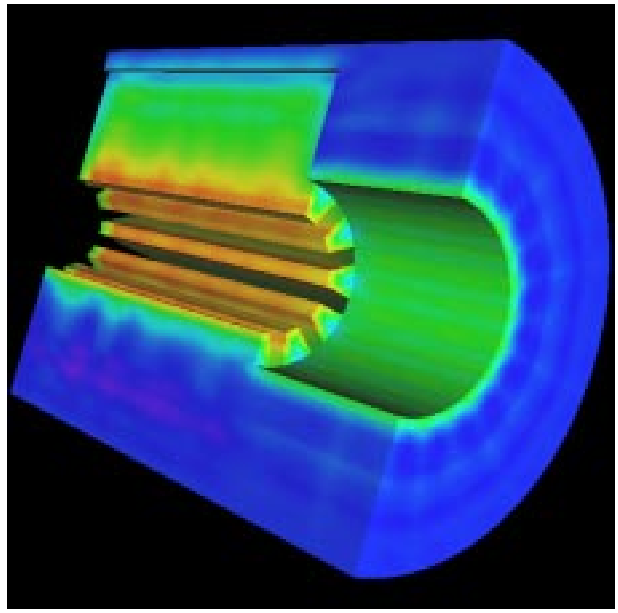

Fig. 3.1.3. Full 3-D simulation of star grain in Space Shuttle RSRM showing stress in propellant due to coupled fluid pressure in core region of rocket motor. Executed on 256-processor SGI Origin2000, visualized with *Rocketeer*.

Combustion and Energetic Materials:

- Homogeneous surface combustion
- Combustion complete in combustion layer
- Combustion layer treated as infinitesimally thin
- Jump conditions that satisfy conservation of mass and momentum
- Pressure dependent regression rate

Computer Science:

- C++ interface code transfers data between fluids and structures, in parallel
- Performance results and scalability studies on local and ASCI machines

System Integration:

- Full aeroelastic coupling of fluid and solid
- Loose coupling (i.e., time steps taken separately in fluids and structures)
- Various coupling schemes implemented and compared (first/second order, serial/parallel, etc.)