verification methods, which employ more comprehensive analysis, but require a great deal of expertise on the part of the user. FLAVERS employs data flow analysis techniques that are efficient and easy to use and are applicable to both sequential and concurrent programs. With FLAVERS, users can specify a program behavior that is of particular importance and then direct FLAVERS to determine whether the behavior will occur on either all, some, or none of the program's executions. For example, FLAVERS can be used to verify safety conditions by demonstrating that a user-specified unsafe behavior cannot possibly occur. Or, it can help with debugging by identifying situations where a dangerous behavior can occur.

The FLAVERS demonstration will analyze programs written in Ada and Java and will highlight a new, more powerful and easy-to-use user interface. A key addition to this user interface is support for browsing paths along which errors can occur.

Formal Alternative Management Integrating Logical Inference and Rationales (FAMILIAR): http://www.kevol.com/KEI-6p.html
Knowledge Evolution/Synquiry Technologies - Dr. Sidney Bailin/Dr. Dean Allemang

The FAMILIAR tool provides disciplined support for collaborative problem solving such as planning or software design. Members of a team use the FAMILIAR visualization modes to examine past cases, construct new solutions, trade off alternatives, and perform "what if" analyses. FAMILIAR includes formal analysis functions that issue advice to help guide the construction and evaluation of solutions.

FAMILIAR allows someone who is facing a complex planning problem to organize the contingencies and alternative plans in a systematic way. It supports reactive re-planning in the face of unforeseen plan failures, by identifying opportunities to re-use plan components (even from plans that were originally rejected). This improves the survivability and fault-tolerance of the plans.

FAMILIAR models problems and their solutions in terms of goals, alternatives, features, and components. Goals represent the objectives of the current task. The alternatives hierarchy is a categorization of known solutions to known problem types. It is a corporate memory of best practice, which may be consulted and borrowed from when confronting new problems. Features are the dimensions along which alternatives differ from each other. Because any complete solution design is a combination of many aspects, the feature hierarchy provides a way to compare and contrast specific aspects of alternative solutions, and to perform "what if" analyses by changing some aspects while keeping others constant.

A solution may be composed of several components, reflecting a decomposition of the problem into smaller, simpler ones (divide and conquer). Of course, choices made concerning the solution to a sub-problem will have an impact on the overall solution. For example, the selection of a particular approach to solving part of the problem may invalidate using certain approaches for other parts. FAMILIAR maintains these dependencies, propagating decisions so that the overall solution remains consistent, and informing the user about the implications of choices made. These implications go beyond a simple yes-no answer as to whether the solution will work. They include tradeoffs indicating how well different goals are met using different solution approaches, and advice on how to fix problems that FAMILIAR has discovered.

FAMILIAR allows a problem-solving team to keep track of several candidate solutions. As alternative solutions are composed, FAMILIAR tracks the rationale for the parts of each solution. It allows team members to evaluate the fault-tolerance of each solution, and to make use of alternative components (either new components or components of other candidate solutions) to address possible failures in the current solution design. In particular, FAMILIAR supports evolution by identifying those parts of the solution that are affected by a change in specifications or operating environment.

FAMILIAR may be applied to a wide range of problem-solving situations. It is particularly useful in situations that are both multi- dimensional (many aspects to the problem) and multi-level (must be decomposed into sub-problems). It is the only decision support technology that manages the interactions between these two sources of complexity on behalf of the decision-maker. As such, it is particularly useful for software design and for planning situations in which there is uncertain, incomplete, or rapidly changing information.

Incremental Constraint Engine: http://www.htc.honeywell.com/projects/dssa/
Honeywell Technology Center – Steve Vestal

The Incremental Constraint Engine provides efficient incremental entry of certain classes of constraints, together with a rapid assessment of feasibility and (in the case of unfeasibility) identification of culprit constraint sets. We have a prototype integration of this capability with DoME to support management of constraints that span multiple models of a system.

Specific demonstrations may include:

- Army AMCOM has created a generic or reference software architecture for the missile domain. This architecture has been captured in MetaH, and the MetaH toolset has been used to analyze the schedulability of the system and produce real-time executables for a variety of target hardware configurations.
- With assistance from Boeing and the Comanche PO, we developed a preliminary MetaH specification for the Comanche Mission Equipment Package avionics. This specification was used to develop a system schedule and perform schedulability analysis.

Related demonstrations may include:

- CMU/Lockheed-Martin have captured a version of their Simplex architecture in MetaH.
- SEI has developed a translator between MetaH and ACME.
- University of Colorado is working to apply their impact analysis to architectures specified in MetaH.

Honeywell has developed a model of portions of the MetaH real-time executive using the MCC/U Mass FLAVERS formal verification toolset.

INSERT - Incremental Software Evolution for Real-Time Systems:
http://www.cs.cmu.edu/afs/cs.cmu.edu/Web/Groups/real-time/insert/
Carnegie Mellon University, Software Engineering Institute and Lockheed Martin Tactical Aircraft Systems - John Lehoczky

The INSERT technology package permits the easy and reliable insertion of new or upgraded capabilities into mission critical systems. This package creates a structured environment based around innovative uses of advanced technologies including analytic redundancy, dynamic component binding, dependency tracking, and data fusion integrity processes. The INSERT approach leads to reduced development and test time, reduced cost, and a reduction of the technical risks involved with system evolution.

The EDCS-INSERT demonstration highlights the application of these tools and capabilities within the context of an F-16 mission system suite. We will be presenting a videotape of a capability upgrade to the F-16 mission system. The demonstration takes place in a ground-based simulation of the actual F-16.

The upgrade involves the insertion of an automated Air-to-Ground weapon delivery capability. During execution of the new capability, a sequence of different failure occurrences will be injected; however, the INSERT architecture will be able to cope with these failures and maintain stable system operation. This capability promises to increase the effectiveness and survivability of the aircraft.

The INSERT demonstration highlights the following:

- The INSERT approach is effective on DoD scale problems. The design properties are relevant and resource efficient.
- The architecture is cost effective. Adoption costs are small.
- The work has resulted in additional architectural approaches that are applicable to a wide range of mission system domains.

The On-Site Demonstrations of Major Components of the INSERT Package include:

- Reliable Upgrade Environments: INSERT F-16 Autoguidance Applications

In addition to the videotape presentation, we are presenting live demonstrations of INSERT capabilities in pure virtual simulation environments. Two demonstration environments will be shown based on an existing USAF/Lockheed Martin F-16 simulator. One environment demonstrates an autopilot upgrade environment designed and generated using Honeywell MetaH on Windows NT 4.0. The use of MetaH (an EDCS technology) supports cross-platform environments and additional architectural analysis. The other environment uses custom code on a Real Time Operating System to provide similar capabilities. Pilot-Relief and automated landing modes are demonstrated.

- Computer-aided Support Tools for Verification of INSERT Switching Rules

INSERT switching rules provide protection against semantic faults that could be introduced in the software upgrade process. The performance of the switching rules over the entire range of possible operating states can be verified using a new tool for modeling and computer-aided verification of hybrid dynamic systems. The capabilities of this tool will be illustrated for the INSERT F-16 autopilot demonstration system. The demonstration includes the building of the model of the hybrid system dynamics, the evaluation of the results of verification queries, and the use of automata approximations to the hybrid system dynamics.

- Semantic Dependency Analysis Tool

Application errors due to hidden side effects are addressed through design-time analysis of a system model. By documenting assumptions about semantic and time-sensitive characteristics of components, the system model allows identification of violated assumptions, inconsistent INSERT configurations, and the impact of the proposed changes. The analysis tool is implemented on ACME/ARMANI, an EDCS technology.

Internet-Based Information Management Technology:
http://www.psl.cs.columbia.edu/current.html
Columbia University, Dept. of Computer Science - Prof. Gail E. Kaiser

Columbia University will demonstrate its Internet-based information management technology for multi-organization collaborative work. Applications are not limited to very large systems engineering, e.g., open-source software and multiple (sub) contractor projects, but also include decision support and distance learning. The main components and toolkits include: Worklets, a mobile agents approach to meta-workflow for dynamic reconfiguration and knowledge propagation; Workgroup Cache, for zero-latency knowledge propagation among dynamically organized groups according to task-specific criteria; Xanth, an XML-based data fusion service; Groupspace Controller, an object/event broker featuring vetoable events and wraparound service activation; TreatyMaker, a toolkit for rapidly constructing and dynamically managing N-ary interoperable alliances among peer services and systems;

JPernLite, transaction management middleware supporting plugin extended transaction models, e.g., for groupwork and "what if" transactions; and TaskWeb, an open hypermedia system for PDA's.

These and other technologies are integrated in CHIME, Columbia Hypermedia IMersion Environment, a framework for generating and managing MUD-like 3D virtual worlds for collaborative information understanding and interaction. Demo scenarios will include software development and multi-agency emergency response.

Jakarta Tool Suite (JTS):
http://www.cs.utexas.edu/users/schwartz/proj.htm
University of Texas at Austin - Don Batory