The Jakarta Tool Suite (JTS) is a set of Java-based tools for developing product-line architectures, application generators, and compilers for domain-specific languages.

JTS is being used in to develop the next generation of FSATS (Fire Support Automated Test System) that has been developed by the University of Texas Applied Research Laboratories. Unlike its predecessor, the new system, FSATS99, is highly extensible: a product-line of FSATS99 simulators can be assembled from components using JTS. Building C2 simulators in this manner appears to have significantly simplified software development, understanding, maintenance, and evolvability of the system. We will demonstrate our current prototype of FSATS99, along with other capabilities of JTS.

Knowledge Depot: http://www.ics.uci.edu/pub/edcs/
University of California, Irvine – Richard Taylor/David Redmiles

Knowledge Depot supports Project Awareness by capturing documents and communications relevant to a project and, based on the interests people register with the system, redistributes summaries of this information to interested people. This approach allows a user to specify what kinds of information affects his or her work and skim through summaries of this information to determine if any of the documents or communications have a potential impact upon their work. A user can not only be alerted when there is a change in a component (or other aspect of a project) that he or she depends upon, but can become aware of the fact that the change is being discussed. Project awareness enables the person to prepare for change, and to contribute to the discussion from the perspective of how the change will affect that individual's work.

Little-JIL: http://laser.cs.umass.edu/perptest/
University of Massachusetts at Amherst - Leon Osterweil/Lori A. Clarke

We will demonstrate Little-JIL, a graphical language for specifying the execution of complex processes by teams of agents that consist of both humans and machines. Little-JIL can be thought of as a multi-agent coordination system, or as a process execution system, that offers substantial enhancements over typical workflow systems, especially in its powerful facilities for handling exceptions, its comprehensive treatment of resources, and the uniform way in which it treats humans and automated agents. Little-JIL process specifications are hierarchical decompositions of steps, where steps are guarded by prerequisites and postrequisites, whose violates throw exceptions. Steps also incorporate data flow specifications, as well as specifications of resource types that are used as the basis for dynamic scheduling of resources.

The demonstration will show the use of the Visual-JIL graphical editor for developing Little-JIL process descriptions, as well as the Juliette environment for supporting execution of Little-JIL processes on a distributed platform of workstations. As part of the demonstration of Juliette we will focus on our resource management system and our Grapevine agenda management generation system. Example Little-JIL processes to be demonstrated will include processes for multi-user software design, multiagent negotiation, and perpetual testing.

Maude:
http://www-formal.stanford.edu/clt/ArpaActive/summary.html
SRI/Stanford University - Jose Meseguer/Carolyn Talcott

SRI/Stanford University will demonstrate formal interoperability of ADLs using the Maude tool. Maude is a high-performance reflective language and system supporting both equational and rewriting logic specification and programming for a wide range of applications. Maude can be used to build executable formal models of system architectures very quickly, at diffferent levels of abstraction and amenable to a wide range of static and runtime analyses including model checking, symbolic simulation, monitoring, and theorem proving.

Model-Based (Systems) Architecting and Software Engineering (MBASE): http://sunset.usc.edu/
University of Southern California Center for Software Engineering (USC/CSE), The Aerospace Corp., and TRW - Barry Boehm/Neno Medvidovic

MBASE is a set of guidelines that describe software engineering techniques for the creation and integration of development models for a software project. The models to be integrated extend beyond Product (development) models such as object oriented analysis and design models and traditional requirements models, to include Process models such as lifecycle and risk models, Property models such as cost and schedule, and most notably Success models such as business-case analysis and stakeholder win-win. The approach used in MBASE ensures that a project's success, product, process and property models are consistent and well integrated. MBASE core model frameworks guide the project's convergence on a consistent and feasible set of models, and guide the product's development or enhancement through an extension to the original Spiral Model. MBASE is highly compatible with Rational's Unified Software Development Process, which has adopted the MBASE anchor point milestones (MBASE has adopted Rational's Inception/Elaboration/Construction/Transition phase definitions for the activities between the milestones) MBASE is trying to extend Rational-USDP'S architecture-centric use case-driven process toward a process which is both architecture-and stakeholder-centric, and both use-case-and business case-driven. MBASE provides a constructive approach and extensible framework of collaborative tools, enabling a system's stakeholders to rapidly develop mutually satisfactory (win-win software system solutions.

The tool framework includes the USC Center for Software Engineering's WinWin, COCOMO II, Architecture Attribute Analysis Aid (A4), and Distributed Collaboration and Prioritization Tool (DCPT). It has been integrated with such other DARPA tools as Rapide (Stanford), C2 (UCI), ScenIC (Ga.Tech), MediaDoc (USC-ISI), JWatch (Intermetrics), and Catalyst (MO); an Aerospace Corp. trajectory simulation and visualization program; and with such commercial tools as Rational Rose, CUSeeMe, and RealPlayer.

MBASE and its tools have been applied on over 50 digital library projects at USC. Early adopters of MBASE capabilities include the Air Force C2ISR Center, FAA, The Aerospace Corp., TRW, Litton, and Xerox. The demonstration will show how these capabilities can be applied to a DoD quick response mission requiring not only rapid mobilization and deployment, but also rapid software change coordination and implementation.

MediaDoc: Automated Generation of Multimedia Explanatory Presentations
http://www.isi.edu/isd/I-DOC/media-doc.html
University of Southern California/Information Sciences Institute (USC/ISI) - Lewis Johnson & Stacy Marsella

Many organizations face a constant problem of obtaining accurate, relevant information about complex, evolving systems. We see this with deployed hardware and software systems, for example. Such systems often have long lifetimes, while staff turns over frequently. New staff assigned to tasks such as maintenance and upgrades have difficulty obtaining the information that they need to perform their specific tasks. They waste time sifting through irrelevant information in voluminous documents, and the information that they find may be out of date. Tactical decision-makers in the military face similar problems, but on compressed time scales. Command staff members need to obtain focused views of an involving tactical situation, and their jobs can be hampered if the information that they receive is cluttered with irrelevant material or is out of date.

The MediaDoc project addresses these problems through the automated generation of focused multimedia presentations of multidimensional information.

A user performing a particular task can pose queries about a system; MediaDoc extracts information relevant to the query and the task, and automatically generates presentations combining text and graphics. Automated text extraction tools are also provided that extract semantic information from relevant textual documents, so that this information can be integrated into the presentations. We will demonstrate how these techniques may be applied to a software engineering task and to crisis management in a military relief operation.

MetaH: http://www.htc.honeywell.com/metah.
Honeywell Technology Center – Steve Vestal

MetaH is an architecture description language and toolset for avionics and other real-time applications. The toolset includes a software/hardware binder, schedulability modeling and analysis, reliability modeling and analysis, and partition impact modeling and analysis. There is also an automated composition tool that builds a real-time executable from an architectural specification and a set of software components.

Model Integrated Computing (MIC):
http://www.isis.vanderbilt.edu/
Vanderbilt University/Institute for Software-Integrated Systems - Gabor Karsai

The MIC Environment developed at Vanderbilt/ISIS will be demonstrated through an application: the Integrated Test Information System (ITIS). IT IS has been developed for and is being used by Arnold Engineering Development Center (AEDC), Arnold AFB in support of engine and airframe ground testing. The ITIS integrates many facility-wide legacy data systems and databases, facilitates user-defined analysis of the information, and delivers resulting information to a geographically distributed set of end-users.

The MIC technology has been used on two levels:

- On the "meta" level, the concepts and semantics of the application domain, and their mapping into the implementation domain have been captured using the meta-level modeling environment (which is just another instance of the MIC architecture). This was followed by the synthesis of a domain-specific modeling and program generation environment, and the development of the run-time support system for the application.
- On the "domain" level, end-users use the environment to develop domain models that are then used in the synthesis of the actual test-specific ISIS application.

The DARPA EDCS technology provides the support for meta-level modeling of the domain, the generation/synthesis of the domain-specific environment, the infrastructure for the domain-specific environment, and the generation technology for synthesizing the application from domain models. The demonstration also integrates other EDCS technologies. For instance, ACME architecture models are generated, which then can be analyzed using ACME tools.

The demonstration shows the meta-modeling environment, the domain-modeling environment, and the generated application. The demonstration also shows how a small-scale, but fully functional ITIS system can be customized on the fly using the meta-level technology.

To show cooperation with other DARPA programs and the dissemination of the technology, another demonstration shows how a domain-specific modeling and analysis environment can be built (and used), for Adaptive Computing Systems (ACSs). In ACSs, dynamically configurable (and re-configurable) hardware and software architectures are modeled, design alternatives analyzed and are explored, and application hardware and software are synthesized.

ORBIT/VIRTUE - Collaboration and Visualization Support for Complex Systems Evolution:
http://www.dstc.edu.au/wOrlds/
University of Illinois/University of Queensland – Daniel Reed/Simon Kaplan

This demonstration illustrates the integration of desktop and virtual environment collaboration systems and their application to complex problems, such as software engineering, logistics and data analysis. The Habanero and Orbit desktop systems support extensible, varying intensity collaboration and tool sharing across local and wide area networks and are coupled via shared controls and streaming audio/video to the Virtue virtual environment for