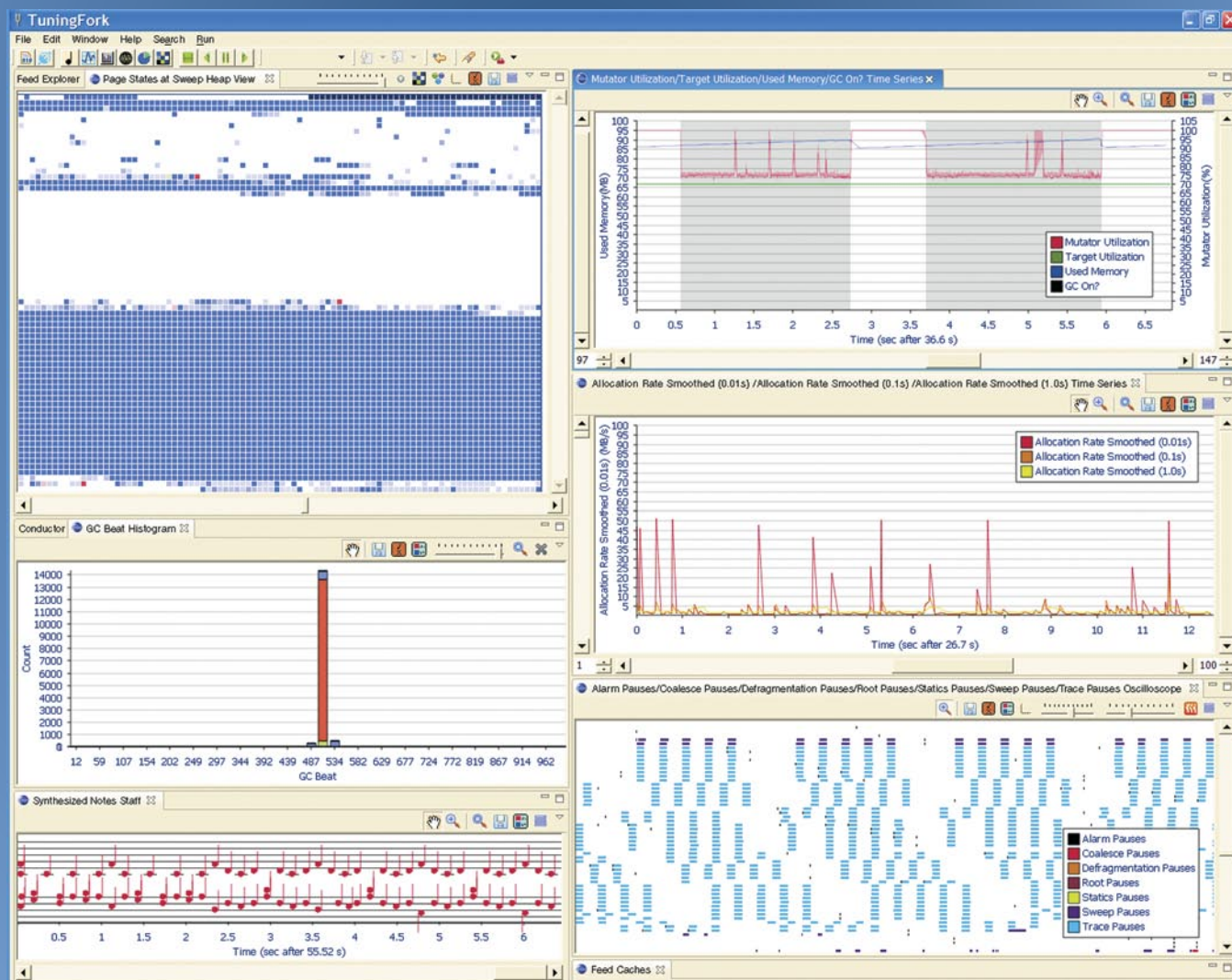




IBM Research has a prestigious history of achievement in programming languages and software engineering that includes the creation of FORTRAN, the development of APL (A Programming Language), pioneering work in the field of compiler optimization, the development of design patterns for object-oriented software, and the invention of the field of subject-oriented programming and multi-dimensional separation of concerns. Spanning the spectrum of software development, from requirements to systems maintenance, IBM's research addresses many issues, such as supporting the development of complex applications with modeling techniques, enhancing runtime technologies with new features, and developing new technologies to ensure that software applications meet the required standards of quality. These projects share common objectives: working with and creating open standards to advance the field of software development, as well as enabling the construction of high-quality, cost-effective, multi-platform software. The impact of our work is seen in IBM software products, as well as in processes used within IBM and in client engagements.

Visualization of a Java music synthesizer running on the Metronome real-time garbage collector.



MODEL-DRIVEN DEVELOPMENT AND INTEGRATED SOFTWARE ENGINEERING

Model-driven software development not only holds great promise for dealing with increasingly complex software systems, but also presents many challenges. These include simplifying model elicitation, defining precise and appropriate semantics for models at high levels of abstraction, and developing tools that both support the modeling paradigm and provide value to the developer teams. Our model-driven development technologies are united by the vision of integrated software engineering, which provides a common framework for our modeling tools. The open-source Eclipse platform, to which IBM Research has provided significant contributions, serves as a foundation for this integration.

Technologies that span the software lifecycle have been developed to address the challenge of building complex systems that are flexible, cost-effective, and maintainable. In the requirements and architecture space, novel techniques have been devised for capturing informal information and for providing structure and rigor through transformations and refinements. Given a structured, rigorous model, our techniques can analyze, execute, and debug it, as well as provide visual feedback to the user before the system is actually constructed. Finally, new validation methods using fault models and artificial intelligence planning for test case generation provide a means to validate that a working system conforms to the original models.

Unique approaches have also been developed for automatic mining of logical data models, behavioral models, and business rules from legacy applications. The extracted models can be used to better understand these applications, to assess the impact of planned changes, and to enable the automation of select code transformations.

NEW MODELS, PARADIGMS, AND IMPLEMENTATIONS FOR PROGRAMMING LANGUAGES

IBM's research in programming languages ranges from design and semantics to runtimes and virtual machine technology. In language design, advanced constructs and technologies are being created to address a variety of programming challenges, including end-user development and programming to allow non-technical clients to easily Web-enable their businesses, treating XML data as a first class element in Java™, and providing advanced support for high-productivity programming on IBM's high-performance parallel architectures.

IBM Research has a record of innovation in the runtime area. In a recent major advance, the world's first real-time system for automated garbage collection in memory-managed systems has been developed. This work provides a solution to a problem that was considered to be intractable, and provides the basis for using Java in real-time systems. Researchers are also using cloneable virtual machine technology to address performance. This allows Java Virtual Machines to be frozen and cloned, reducing or eliminating expensive startup steps.

PROGRAM ANALYSIS AND MEASUREMENT FOR SOFTWARE QUALITY

In the software quality area, IBM Research's focus is on using automated analyses to find defects and problematic programming constructs. These techniques allow users to pinpoint problems in an application. Researchers have developed pioneering dynamic analysis technology for detecting problems based on program heaps. This technology is able to identify memory leaks and performance bottlenecks for heaps in excess of 20 million objects. The detection of errors and anti-patterns is also being addressed by using static analysis to check programs for properties specified by the user, as well as to check for conformance between specifications and the program being analyzed. Both static and dynamic analysis-based approaches have been incorporated into IBM software development platforms to help clients produce high-quality applications.