

001 Testimonials

Professor Michael Golay and Meng Ouyang
Excerpt from MIT report An Integrated Formal Approach for Developing High Quality Software of Safety-Critical System, Report No. MIT-ANP-TR-035

"An extensive study has been performed concerning currently available formal methods, design approaches, and available supporting tools...By following the selection criteria that are based upon actual industrial application requirements, the formal method named Development Before The Fact (DBTF) is selected for use in this study. This method has a sound mathematical formality and a very powerful supporting tool: The 001 Tool Suite...The most valuable result from this research is that the selected DBTF formal method, its supporting tool -- The 001 Tool Suite, and the proposed Integrated Formal Approach prove to be useful in a practical industrial project right now for improving the current development of safety-critical system software."

Dick Spencer
CEO, Virtually Perfect Software

"Despite recent progress in object oriented programming and component-based software engineering, software development today is still more an art than an engineering discipline compared to chemical and computer hardware engineering. HTI's 001 Tool Suite is a real breakthrough, because of its ability to transform today's software development craft- oriented industry into an engineering discipline.

Our world's global economy demands increasingly complex information systems to provide us all with more prosperity, security, and control over our environment. Unfortunately, the demands for these complex information systems depend upon the development of more capable software systems, which results in increased growth in functionality and, hence, complexity. As software complexity increases, so does the probability of errors, which may cause a catastrophic loss of property, time, and even human life. The prime virtue of the 001 Tool Suite is that it confronts complexity by relying on a mathematical or formal methods foundation to enable it to produce virtually perfect software by the automated elimination of defects. 001 developed software is not absolutely perfect, because it can't remove 'user intent' errors, such as an error in implementing an algorithm or a requirement. However, it does

produce virtually perfect software, because it automatically eliminates virtually all internal errors and inconsistencies. We need the engineering discipline facilitated by HTI's 001 Tool Suite to develop the dependable information systems required for our future prosperity and security."

Michael Huang
Technical Demo, Ariel Technologies

"Most of the popular programming languages (i.e., Java, C++) have nice object oriented features to help programmers make software. However, they don't have any mechanisms to force them to use object oriented features. For example, one could use Java or C++ to make his/her software without using any objects in the programs. Therefore, since there is no forced mechanisms to adhere to standard processes in these languages, oftentimes developers can still have the problems identified above.

001 programs automatically contain reusability and flexibility. Therefore it is easy for us to modify them. For this reason, in one of our programs, it took us more than half a day to make a change in Java. However it took only a couple of minutes to make that same change in a 001 program which did the same thing."

Tom Rona
Former Acting Science Advisor to President George Bush

"001 systems have built-in quality and built-in productivity."

Development Before the Fact Feasibility Study
Toldark Pty. Limited, Canberra, Australia

"Most of the systems development technologies available today be they based on specific database management systems or whatever, are unable to link the application requirements definition to the final product. Therefore they are curative; that is, they start with many errors which must be eliminated manually one by one at great cost over the product's life cycle.

Preventative development technologies introduce the concept of preventing errors from entering the development process. Achieving this goal transports software development from the artistic into the engineering domain. The resulting consequences will be major increases in product life expectancy, the elimination of constant curative activities, and partial or total reusability when faced with functionality changes. The capital spent fixing errors can now be diverted to productive activities.

The wide-scale use of a preventative software development technology

is likely to impact world economics to an extent comparable to the appearance of the transistor. In more specific terms and on a shorter time scale, the impact of reliable corporate systems benefiting from amortization periods equal to or greater than any other item of plant and equipment would enable world economic benefits of immeasurable magnitude.

Tools capable of achieving this formidable step will gain acceptance and forever change the software industry.

Development Before the Fact from Hamilton Technologies Inc. in Massachusetts in the USA is the most promising of the software development technologies of the future that has proven itself in various environments...it could very well become the leading software development methodology for years to come.

The Development Before the Fact methodology is driven by the formal definition of requirements which drives the design process thus preventing errors. Therefore this technology is preventative. A constant market vigil has been carried out and every claim implying preventative development capability has been checked. To date, we have not encountered any other product which changes software development from a creative art to a scientific activity; i.e., the results are mathematically provable. Testimonials over many years have been accumulated. "

Hal Camp
Senior Engineer and Site Manager, SAIC, Washington, D.C.

"There is a significant need for a complete computer based system engineering environment. When I spent a month studying the 001 tool suite I found such an environment. Over the past three years I have come to understand that HTI has created a path which, if followed, will alter the software culture by creating a true engineering discipline."

Larry Bowe
Bowe Software Solutions

"The unique qualities of 001 are it's ability to support the entire software development process from requirements definition to field maintenance. There are facilities that support requirements definition and management, software architecture design, automatic document creation, automatic code generation in multiple languages, requirements based testing, software code coverage testing, regression testing, project management such as process and progress measurements, and configuration control. No other tool that I am aware of is so complete and

encompassing. At best other solutions require use of multiple tools that are not tightly integrated, which causes the developer to learn multiple user interfaces and struggle to overcome integration compatibility issues. This tightly integrated solution should result in a much higher return on your tool investment dollar."

Bernard Antonuk
Praxis International

"Development before the fact as implemented by 001 has proven to be the only automated methodology currently available on the market that will allow you to verifiably model and generate (or model and implement) any type system in any organization in any industry in any type of market...Hamilton's work stands to be to software engineering what Einstein's work was to physics."

Wayne Painter
Pacific States Marine Fisheries Commission

"The 001 Tool Suite is truly a fully integrated and complete environment for developing complex systems. The developer uses one tool set, one environment and one language to completely analyze, design, specify, program, integrate, test, install, execute, document and maintain highly complex software applications. Application modules are fully integrated when the design is completed. It is the only tool set I know of that actually generates 100% completely executable applications. Without 001, I estimate it will take 5 to 10 times longer to develop and complete a project and the results will not be nearly as dramatic or powerful.

I know that the Development Before the Fact method of defining and development systems is by far and unquestionably the best method I know of; it is very similar to one I evolved into in my own personal experience in developing software development tools. But HTI's whole approach has gone much further and most importantly, it is based on mathematical foundations. That is why I chose HTI's 001 Tool Suite as my primary design tool to develop a highly complex Fisheries Information Network in Alaska. Once you begin to use 001 you will realize that it has awesome power and ability. It is a truly revolutionary way to finally do real software engineering. I applaud Margaret Hamilton and her team of dedicated workers in developing one of the most important design tools of this century. "

Ernie Pearce
President, Western Colorado Med Net

"While at BDM Technologies as Vice President of Information Technology, I had the opportunity to become very familiar with 001. I led

teams that produced two demonstration products for clients. We developed an enterprise level model for Circus-Circus Enterprises, Inc. and drove the function for real estate development of new casinos down two additional levels of detail. We also built an executable program (a customer support help desk application) as a demonstration of the speed and effectiveness of 001 for the sales division of MCI. We completed this product in under two weeks. I have worked with dozens of systems through the years and have never seen such a powerful and easy-to-use system as 001.

Thomas J. Harris
Software Development Manager and Consultant

"Having been involved with Software Engineering for nearly three decades, I have seen the advent of many new technologies for the practitioner community. With the ability to span modeling through-development, 001 is a major advance for software designers and developers. It's the first lifecycle tool that actually delivers on productivity gains throughout the course of a project. It's also the first to be sufficiently robust that I would not hesitate to recommend it for large or mission-critical applications."

Earl L. Santee
Sacramento Sit Director, Access Research Corporation

"Until 001, our company had relied on structured analysis techniques (SAT) to perform verification and validation on our customer products. Illustrating specific aspects of any system using SAT was a great step forward, however, 001 provides design traceability, and validation before implementation. For an IV&V activity, that means a smaller group of design analysts can be in a better position to more quickly challenge the prime design teams. The IV&V team has unprecedented power to point out graphical, design requirement deviations and to test the significance of those deviations before any coding is ever attempted. The ability of the HTI tool set to combine the power of OOP design with mathematically proven control and data structures to produce seamless Type and Function Maps of system requirements is, in my opinion, revolutionary. The HTI tool set provides us with technical leverage and greatly increased confidence that we can trace system requirements to any desired degree of detail."

Taken from: Distributed Discrete Event Simulation using 001 by Roozbeh Ghorishi and Avanish Sahai, Boston University, Boston, Mass., published in the Annual Symposium on Parallel Processing

"001 has proved to be a very efficient development environment. It

allows the user to implement and simulate logically correct systems with a high productivity rate by automatically performing various tests and analyses which are normally the programmer's responsibility...the parallel algorithms."

Fred Robertshaw
Facilities and Capital Manager, Rexham Aerospace and Defense Group Huntsville, Alabama

"We selected the Hamilton 001 tool as our primary design and development tool for RADG 's Advanced Flexible Manufacturing Facility. It was the only available tool which could model the entire factory operation prior to construction."

Keizo Watanabe
NRI & NCC Co, Ltd., Shinjuku-Ku, Tokyo, Japan

"We spent a lot of time before we found this powerful tool that could be applied to our complex, large-scale systems development in the real world such as the securities market. 001 is the most integrated and sophisticated. I believe it's the way to go."

Andy Blum
Publisher, CASEWorld

"I can see no real difference between real-time systems OLTP and business solutions developed using modern technology exemplified by 001. No real-time system should be without it..."

Max Schindler
Computer-Aided Software Design, John Wiley & Sons

"It (001) makes sure that from the very beginning system definitions exclude ambiguities and that module interfaces can't clash...a miraculous methodology..."

Edward J. Lanigan
President, The Lanigan Group, Inc

"HTI's 001 software development tool suite represents one of the few truly revolutionary products available on the market today. Prior to evaluating any tool I've found one question to be the most telling. I ask 'do the system and software developers of your tool use the tool in the development process?' With one exception, not a single vendor answered yes to this question and was prepared to prove it. The second question then becomes 'why should I invest in your development

approach when you have not'. The one exception is 001. Not only is the product used in the development process, but the production code that is the delivered product has been autogenerated by the Resource Allocation Tool (RAT) portion of the tool suite. Embarrassing questions aside, the truly significant aspect of the 001 tool suite is the development approach that it automates. While other tools automate a portion of the 'well established software development process', Margaret Hamilton and her team have taken a more global look at the problem of software design. As any seasoned engineer studying a bad product would do, HTI evaluated 'software' and 'the software design process', isolated problems and engineered the solution.

Mike S. Bailey
Senior Systems Engineer, Independent Engineering, Inc.
on contract to the Army Test & Evaluation

"We were looking for a tool which could provide a common architecture and maintenance environment for a vast array of applications used in software process and product management. In our search we encountered tools which dealt with one or two elements of our problem, but aside from 001 there was no other tool that could support our entire development and maintenance requirements. During my training at Hamilton Technologies I personally witnessed how 001 ensures the adaptation of a foolproof specification for an application, which, in turn, leads to a verifiable design and production ready code based on that design. In my opinion 001 is not only a CASE tool, but an entire software engineering environment."

Steve Dolha
President, Cadeon Strategic Technologies Inc.- Calgary,
Canada

"In my opinion 001 is light years ahead of any other product on the market. We spent a lot of time looking at several of the popular environments and were constantly disappointed until we discovered 001."

"001 and the "Development Before the Fact" (DBTF) methodology were the clear winners after we completed an intensive six month investigation of current OO methodologies and software development environments. In my opinion, Hamilton Technologies provides the best tool and methodology for building large, complex and error free systems in any business domain. Any IT strategic plan MUST include 001 and DBTF to successfully move beyond the software development crisis of missed deadlines, blown budgets and software that can not evolve rapidly to meet changing business needs."

After using 001 for two years, we've seen 4-10X productivity gains achieved using 001 over traditional hand coding techniques--and by analysts who can learn to use 001 effectively within 1 month. And this productivity increases dramatically (orders of magnitude) as analysts learn and are able to apply reuse techniques to their models from a library of reusable 001 USL patterns.

We truly believe that evolutionary software development is viable using 001, even with the rapid turnarounds of systems changes required by changes in the business and even when the starting point is a legacy system."

Jessica Keyes

Handbook of Technology in Financial Services, McGraw-Hill

"There's a methodology out there called Development Before the Fact that's not only going to knock your socks off--it's going to radically alter the way you develop financial services applications--to the tune of fewer programmers, lower costs, and more correct systems."

Excerpt from article Emerging Technologies Keep Development Dynamic, published in Application Development Trends Magazine

"Ultimately, the tools of the future will enable users to run their systems on any platform, under any operating system and across any database--even the Internet. The architecture will be open, the development environment completely visual...the one tool that does that today, 001, is out of the technology lab at Cambridge, Mass.-based Hamilton Technologies, Inc. It is a complete visual systems engineering and software development toolset that invented the concept of cross-platform heterogeneity. Object- as well as function-oriented, 001 uses the metaphor of type and function maps to sketch out and then simulate a system. Once the design is refined, 001 can generate pretty much error-free code to any platform, running under any operating system, reading any database and accessing any programming language...In today's OO, intelligent, data warehouse, OLAP, Internet-oriented environments, tools such as 001 just might lead the way in providing I/S with a single toolset that can do it al...As Margaret Hamilton, president of Hamilton Technologies, Inc., puts it, "The world is becoming much more complex, New enablers such as the Internet are requiring developers to become more and more creative in solving business problems. But even as this complexity increases, there is no reason why our development

environments can't be organized and rational. There's no reason why a developer shouldn't be able to use a single suite of tools to develop across different technologies."

Excerpt from book Technology Trendlines, published by Van Nostrand Reinhold

"I have been following 001, as well as the rest of the software development industry, for several years now. Although it has not quite happened yet, I predict that where 001 goes, the rest of the industry will follow. It is not only that 001 is so radically different from anything else on the market, it is that the main mission of 001 is to promote unbelievable leaps in productivity. That is really the bottom line...001 permits a systems designer to sit at a workstation with one or more end-users. A system is jointed, graphically designed, and prototyped. After several iterations of this process, the final system is automatically generated as is the documentation. Is a programmer necessary for this? No. What is needed is a team of expert end-users and a systems analyst/knowledge engineering type who understands the business."

001 User Profiles

Advanced Methods

Picture this.

A bank in Basel sends a message to a bank in New York to transfer ten million dollars. Sound simple? But what if the message is in German and the format anything less than readable? And what if the message is less than legit?

Welcome to the complex and high-risk world of banking funds transfer. A world where 001 fits in perfectly.

Frank Middleton is a partner in the Saddle Brook, New Jersey firm of Advanced Methods, a firm specializing in the use of advanced technological solutions to business problems.

Middleton's specialty is in applications design. When one of the largest banks in the world needed to architect a productive and reliable solution to the funds transfer problem, 001 was chosen over other tools.

Using 001 Middleton became practically a one-man development team and ultimately generated a funds transfer system containing some 180,000 lines of C code. Explains Middleton, "The productivity gains that we made in developing this system were fairly spectacular."

Not only did Middleton generate 180,000 lines of C code, he also generated thousands of lines of command language.

This is an important point. For not only did Middleton have to worry about building a new, and highly complex system, he also had to worry about moving this system from its original implementation under VAX VMS to a HP workstation under Unix. Two platforms. Two variations of the system. And two completely different command languages (one of which, Unix, he was not yet fully familiar with).

But because 001 has the ability to re-generate 001-generated code to any hardware platform, operating system, database, graphical system and command language, all Middleton needed to do was "specify and then generate."

It's this same ability that makes maintenance a breeze as well. According to Middleton. "The thing I think is really fantastic about 001 is the maintenance side of it. I am able to add new features and make dramatic changes to the software and have it work straight away without having to do any really difficult work."

The bank's success with 001 comes after a series of misses with other tools. In fact, when this project began five different tools were being prototyped. "At the end the only usable tool was 001," says Middleton.

What did 001 have that these other tools did not? "001 provided a complete functional description, a model of work plus the usable code. These others did not," explains Middleton. "There are three types of tools for development--upper, middle and lower--and 001 is the only one that covers all three levels." he continues.

What Middleton particularly likes is 001's object-orientation. In fact, he thinks that 001 goes well beyond today's OO model. "001 is completely OO. In fact, 001 was probably one of the first OO tools available. It creates pieces of reusable code that are very logical and self-contained."

In fact, Middleton is so excited about 001 that he and his partners are making it an integral component of the Advanced Methods tool suite. Why? As Middleton explains it, "001 is the most comprehensive tool I've seen."

The software Engineering Institute (SEI) at Carnegie Mellon University is perhaps the most influential software technology "think tank" in the country. Located in Pittsburgh, Pennsylvania, SEI is a federally funded research and development center under contract to the U.S. Department of Defense. The objective of SEI is to provide leadership in software engineering and in the transition of new software engineering technology into practice.

And 001 is helping them achieve their goals.

Bob Krut is a member of the technical team that is working in the complex arena of Domain Analysis. According to Krut, "We needed a tool to automate our methodology." So, 001 was chosen to assist in this rather daunting task.

"we used 001 to integrate four very different views of a software system into a single tool, so to speak," explains Krut. Using 001, Krut's team fully integrated a features model, an Entity-Relationship (E-R) Model, a Data Flow model and a Behavioral model.

Given the diversity of these modeling paradigms, SEI's preference was for a tool that provided an integrated solution. "We used 001's TMaps for the Features and the E-R model and the FMaps for the Behavioral and Data Flow models," explains Krut.

Because they're research-oriented, Krut and the SEI had no real need to generate production-oriented, executable code. What was really wanted, and what 001 delivered, was a tool that helped them.

What they also wanted was a proof-of-concept. According to Krut, "People come in and we have a discussion about domain models and most of the time they actually want to see something that works based on our discussions. I have a 001-generated demo that shows, by picking and choosing of certain features, that you are able to represent a particular software system.

To create this prototype Krut "picked just the parts of the 001 tool that I wanted." Even though 001 is fully integrated and works like a seamless engine, users have the luxury of revving up 001 in the manner most suitable for their purposes.

Ultimately the domain that Krut and his team chose to prototype was Army Movement Operations which involves the coordinated movement of supplies and vehicles. According to Krut, "We gathered the information that we needed to have in the model." And 001 was the engine that

represented it.

IBM

When the government decided that they wanted to build a National Launch System, they turned to IBM to help them out. But this government contract would be slightly different. Mark Michrina, a System Engineer with IBM in Houston, explains: "They asked IBM to do an analysis of the marketplace for tools that could automate the requirement, analysis and definition components of a project known as the National Launch System. After all of this analysis, and I looked at the major tools in the systems design and development arena, I concluded that 001 is clearly the most powerful design and development tool on the market.

When Michrina went searching for a requirements tool he knew that it would have to be robust enough to handle the rigors of a system with the complexity level of a National Launch System. A Launch Management and Control System (LMCS) is an extremely large and complex system for rocket manufacturing and processing. The goal of the system is to have all developers, which includes all the different NASA, Air Force and contractor sites, linked together on a wide-area network for purposes of sharing data -- and often in real-time, such as downlinking telemetry from spacecraft.

So why 001?

001 offers a precise specification language which the systems engineers can use to accurately, and precisely write down what they want the system to do. "If, rather than writing three volumes of documents that are in English (and ambiguous), a systems engineer can use an exact specification language that is provably correct, then you cut considerable time out of the life cycle. Productivity definitely improves." according to Michrina.

Basically, what Michrina is referring to is integration. One of the major flaws of the current mode of development is that the many different levels of the life cycle beget many different levels of integration problems. In other words, integrating the disparate results of the different life cycle stages into one cohesive framework is fraught with error. Michrina explains, "If a systems engineer (who's responsible for an entire system including hardware and software) is able to accurately specify what is required to the point where a precise specification can be turned over to the developers, then there is absolutely no question as to what it is the developers need to build. The transition between the two levels is

seamless. The integration is built into the process." Michrina goes on to say that if what the developers need to do is to develop software, then 001 has the capacity to generate truly complete and production-ready code. And, "The need to maintain code is eliminated. Problems are addressed at the specification level."

Michrina contends that this is very important: "It's really something that goes beyond what today's traditional design and development tools do." He goes on to say that today's development products are "all bits and pieces," and this is something that his customer does not want. What Michrina's customers **do** want, and the reason he selected 001, is to get things done much faster and at a higher quality level.

So Why 001? "Because the tools that are out there aren't addressing the needs of the community. I think that 001 is hitting it 'head on' more so than any other tool that I've seen."

McDonnell Douglas

Creating a robust design environment is not easy under any circumstances. But when the company is McDonnell Douglas and the target is commercial as well as military avionic systems the task is particularly complex.

But this was the task that Ed Lanigan, then Electronics Unit Chief at McDonnell Douglas, chose to undertake: "We wanted a set of methodologies and tools that would allow us to smoothly evolve all the avionic systems from the customer requirement to the delivery of the product."

But then Lanigan's team heard about 001.

What initially attracted McDonnell Douglas to 001 "was that its approach was very similar to the approach that electronic engineers use. That is, you develop low level reliable systems based upon those component structures," explains Lanigan.

What also attracted Lanigan was 001's object-orientation. According to Lanigan, "There were no object-oriented development tools at the time. 001 was the only one that had taken that philosophy." Reusability is the major thrust behind object-orientation, and 001 certainly adheres to these tenets. However, according to Lanigan 001 goes even further.

"For one thing the design approach from the onset needs to be object-oriented in nature. I think 001 is way ahead of even some of the prophets

out there that are doing object-oriented analysis, " asserts Lanigan. The object-oriented paradigm includes more than just programming. It includes analysis and design as well. But these last two areas are still uncharted territory for most design and development tools. But not for 001. Unlike other tools on the market, according to Lanigan, 001 enables, "the architects of the system to talk the same language as the detail designers."

"Talking the same language" is perhaps one of the more important factors in being able to build and deliver a system. According to Lanigan, "To do this you need to start with a tool, take the product to production, and then maintain your product using the same tool suite that you delivered it with. 001 lets you do that. Other tools make you stop in the middle of the design and change your tools which is not real conducive to high quality code."

Lanigan was dealing with a host of complexities in attempting to build a general design environment which would ultimately be shared by all departments within McDonnell Douglas. Capturing the requirements for a task of this magnitude required a tool that exhibited all of the traits discussed above and then some. The tool would also have to enable requirements traceability, generate everything from a detailed design model to complete, production-ready code to English-language documentation. And it would have to do this all in a productive and reliable manner. Lanigan made the right choice with 001, "I look at system development as part of a big picture. And from the big picture I don't think there's anything out there that can compete with 001."

Today Ed Lanigan is president of the Lanigan Group, Inc.. He has taken his expertise in building real-time, complex systems a step further. He now works in the area of process improvement. And in this role Lanigan has found that, "there are no good tools available to plan for and monitor work in progress. "So on his slate for deliverables is to build a tool for real-time work management. And the tool that he'll use to build it? According to Lanigan, "001, of course."

Lockheed Martin

When ORLANDO-BASED LOCKHEED MARTIN began their evaluation of application development tools they weren't just looking for an ordinary development tool. According to Bob McCauley, "we were looking for a tool that provided a complete portal to portal development environment and, that provided a significant increase in productivity."

One tool that meets this criteria says McCauley, a software manager for

simulation and advanced systems at Lockheed Martin, is 001. The reason? According to McCauley, "With 001 we can start at the requirements level and go right through to turn out code, and it provides significant productivity improvement over traditional methods"

McCauley is using 001 in the area of vehicle simulation which makes good use of 001's ability to automatically generate both ADA code and 2167A documentation.

With 001, a problem is solved in two views--data, and the functions that process the data. The data view and the function view comprise what McCauley refers to as "001's seamless methodology." McCauley explains, "The problem domain starts off with some high level declarations which are decomposed and amplified. They are subjected to a process which provides an evolution from the abstract to the concrete in a continuous fashion."

McCauley often takes the printed output of 001 in the form of function view and data view and pastes them up on a wall. "I get a continuous presentation of the entire functional breakout, or the entire data declaration breakout, which provides an ability to communicate and provide effective documentation," he explains. By seeing a panoramic view of the entire problem solution one is able to have a better, quicker understanding.

But even being able to see a panorama of the entire problem solution doesn't replace the ability to automatically compare requirements against code actually generated. For this McCauley uses 001's Requirements Traceability RT(x) Component. The idea is to be able to account for the satisfaction of customer requirements. "001's Requirements Traceability allows us to match up and account for each requirement as to where those requirements have been allocated in the specification and then trace it down to the code that actually satisfies those requirements." says McCauley.

One result of using 001, according to McCauley, is in the area of maintenance. He gives an example, "in our vehicle simulation area we had a case where a simulation was implemented and then turned over to another programmer. We found that 001's panoramic view of the problem solution has provided a rapid transition which is one of the most important issues in maintenance."

McCauley and Lockheed Martin are representative of the fast changing world of information technology. Where in the past we used the time consuming waterfall model of system development, today's fast pace requires a more rapid spiral approach. According to McCauley, "The key to being able to do this is the ability to accommodate large changes."

Ultimately the tool required to support this model of software development must be, as McCauley has already described it, "portal to portal"--and seamless to boot.

Systems Automation Technology

Even before Jutinder Ryatt and his partner started London-based Systems Automation Technology (SAT), they knew that they needed a solid development methodology. It couldn't be any methodology. It had to be one that did it all.

Out of the many methodologies evaluated, Ryatt and partners found just one that met their demanding requirements. It was a methodology that innately understood how to build reliable systems. It was a methodology that innately understood productivity. And it was a methodology that innately understood reusability. What methodology was it?

Development Before the Fact.

Although Development Before the Fact can certainly stand alone as a development methodology, the fact that it's also part of a suite of systems engineering and software development tools was one of the main factors that attracted Ryatt and SAT to Hamilton Technologies. According to Ryatt, "We felt that the standard development tools did not address the complete software life cycle--whereas 001 does."

Addressing the complete life cycle is extremely important to SAT. As a software development company, primarily in the database arena using Mumps as a programming language, SAT wanted to be able to not only analyze and design their systems, but to generate Mumps code as well. And according to Ryatt, "Most tools don't generate code, especially with the built-in reliability of 001."

When SAT went looking for a methodology/development tool to fit their very exact needs they actually looked at most of the major players in the market. "We looked at the various tools and found them to be fairly disjointed, not integrated. That seems to be true of most tools on the market today," explains Ryatt.

But 001 is different. As Ryatt puts it. "We think that 001 is far ahead of other tools. It's more than just Object Oriented. It's a step ahead. It's, in fact, a radical departure from today's methodologies."

001 has provided SAT developers with a methodology and tool suite for developing, according to Ryatt, "reliable software." SAT has made good

use of 001's prowess by using it as a framework for database development. "We have built tools for real-time database (RTDBMS) applications by incorporating concepts of 'concurrency control protocols' and novel real-time scheduling algorithms," according to Ryatt. SAT has even used 001 to automate the many jobs for building and maintaining a stable database, which database experts refer to as referential integrity.

So why has SAT chosen 001 as its tool of choice in developing the often complex, and always demanding, applications in the database arena? One reason is 001's reliability and productivity, according to Ryatt, "As time goes on, 001 will certainly be looked upon as a standard." But perhaps, even more importantly, is 001's versatility in letting SAT incorporate their own ideas into 001 concepts.

Further 001 Testimonials

The staff at Apogee Communications Technologies

"Apogee Communications Technologies Inc. has used the Hamilton Methodology for many years on a variety of successful projects, and continues to use it today. The first was the design and implementation of the Testword and Message Repair facility of the funds transfer network of one of the world's largest banks. This system, about 300,000 lines of C and 80,000 lines of DCL (all of which was generated from FMaps by the tool), was completed in about 3 years by a team ranging from 2 to 4 people and maintained by one person for many years thereafter. One particular event demonstrated the power of the methodology - the testword datatype had been layered on the (32 bit) integer type. This turned out to be too short for real testwords, and the testword datatype was relayered on a string arithmetic datatype and using the axiomatic testing concept was fully implemented and tested in a couple of days. This system had to provide high reliability and high availability and has done so for many years. Having all of the system described in 001 USL form, including all the operations interfaces and interactions meant that the documentation was always up to date and consistent, which made maintenance (i.e., enhancements)--the methodology allows for a very clear distinction between "user intent" and implementation errors, and the latter tends towards zero very quickly) very easy.

This system (and several others also designed using the methodology) is layered on a proprietary database system widely used in the financial services industry. 001 was used to prototype the use of DCE to separate the application from the database logically and physically. Essentially 001 was used to design and build an ORB, long before CORBA, so that the application could be entirely independent of the actual database

being used.

The next application of the methodology was to prototype a database integration application for one of the world's biggest communications companies. Over many years, different views of the infrastructure (e.g., plumbing, electrical, mechanical, etc.) had been accumulated in a variety of databases (both commercial and proprietary) as well as in AutoCad drawings. An OODBMS was used as a persistent store behind an OO1 designed and built ORB to collect and mediate the disparate views of the infrastructure data, and eventually to repopulate the various databases and drawings with corrected, consistent data. The prototype ended up having about 150,000 lines of generated code and had client components running on many different operating systems.

Another interesting application is the prototyping of a high performance distributed trading system. As time to market becomes increasingly short, it becomes increasingly desirable that prototypes may become the core of a deployable system, and testable architectures can save a great deal of time. The OMap editor makes it possible to instantiate an architecture and try it out very quickly. The interesting parts can be implemented using FMaps and TMaps until the whole system is built. A simple messaging interface allows OO1 to drive a thin client Java UI via a Web server written using OO1.

Many hours have been devoted to discussing how OO1 relates to traditional object oriented paradigms. OO1 is truly Object Oriented by most common definitions. However, it really goes beyond the OO Paradigm".

Dan Davis
Staff Consultant, AGS Nynex

"It doesn't manage the work. It does the work."

Carl Bogenholm
Los Alamos National Laboratory, Los Alamos, New Mexico

"It's my professional opinion that their tool, called OO1, is revolutionary. No other product matches its capabilities, both in what it does and in productivity gains".

The staff at BMC3 Engineering
McDonnell Douglas Space System Company, SDIO
Systems Engineering and Integration Program,
Huntington Beach, California

"When you are evaluating a system/software development methodology ask yourself three questions: 1) Can it truly automate the entire life cycle from specification to code while also increasing productivity and understandability? 2) Does it possess a proof theorem that guarantees logical correctness even for asynchronous real-time and distributed systems? and 3) Is it powerful enough to create an automation tool that can actually generate itself? Only one can answer 'yes' to all of them: 001."

Stan Terry Terry Consulting

Stan Terry, principal of Terry Consulting was tasked by a data-mining software product company to replace their existing tool (which was largely written in COBOL) with C or C++.

Their COBOL code was well suited to the implementation design and had been through several generations of enhancement and bug-fixing. The system Architect of the data-mining system (with whom Stan worked) was very worried that a hand-coded C or C++ version would severely set back the product's reliability. Thus he was open to a 'higher-level' re-implementation.

"The ideal was to be able to enhance the product at the design-level and have the implementation be automatically generated. After looking at a number of tools we concluded that only with 001 could we achieve this.

The criteria established were:

- 1. The tool should generate 100% of the code (no hand-coding)*
- 2. The tool should be implemented in itself (experience shows that a programming tool must itself be used by the implementors to achieve a really robust tool.)*
- 3. The tool should be capable of having its generation capability extended by the using team preferably in C.) This would seem to conflict with requirement 1., but it does not; the extensions become part of generation capability and once tested in isolation can be used with the same assurance as the native classes.*

001 was the only tool that passed these tests."

Norman Beaulieu CEO, Ariel Technologies

"We spent nearly 2 years researching the best development tools available. Don't waste your time; there is nothing even close to the 001 Tool Suite". 001 is very easy to learn and to debug with. Plus, its ability

to generate 100% complete production ready code has saved our developers months of development time and consequently provided us significant competitive advantages. We are continuously amazed."

Bob Frater
Director, CSIRO Institute of Information Science & Engineering, Australia

"I would like to give my impression of Hamilton 001 after reading a lot of material about it and then meeting with Margaret Hamilton. For the first time I can see a software and system design approach that might provide the growth path for software that integrated circuits did for hardware.

Its major contribution to society may well come from its use in helping us model and understand complex systems and organizations."

Phyllis Rye
Rye Associates

"001 is a software development system and methodology which facilitates most, if not all, phases of software development--from systems requirements through implementation. 001 performs generic autocoding--not limited to a small domain of application areas.

001 is unique and powerful. Its capabilities are seamless and end to end, starting from the beginning with process modeling and continuing throughout system engineering and software development. It has both error correction and error prevention capabilities. Most important, 001 has built-in longevity which means that there is now a way for organizations to build systems that will inherently evolve. I know of no other tool (or set of tools) that has these capabilities"

Software Development Director of a Federal Government Agency

"...an opportunity to leap-frog past industry to the cutting edge of software development...."

Terry Ortlieb
Business Development Manager, Global Services Division, IBM

"Things that I have done with DBTF and 001.

Used this technology to verify our implementation methodology. We

were able to validate the process flow and input and output of all of our work products.

Used this technology to define opportunities for parallel processing. These opportunities were in Business Process Reengineering efforts as well as Information technology efforts.

Brought the rigor of engineering to the Business Process Reengineering methods.

Used the technology to assess the validity of philosophical systems. Validated that the process identified by the philosopher transformed some input into an output. Often these philosophical processes were void of any transformation and I was able to discover this easily with the technology presented within this work.

Used the technology to develop middle-ware for managing the interfaces between different package software. The product helped in maintaining the version differences and identify those interfaces that needed to change before the system before it was brought online.

For us at IBM Business Process Reengineering is a science not an art. Object thinking provides the rigorous techniques that put the engineering into Reengineering.

The world is getting more granular. The concept of the world car for the automotive industry is being replaced by customer defined cars. In order for this to work it will be necessary to manage the interfaces between the parts to determine that the customer's concept is possible. DBTF provides the methodology and techniques to manage these interfaces.

Mass customization and enhanced productivity are dependent on object technology. You need to successfully manage the connection of objects and provide a method that can:

- prove the object connection prior to building the object*
- identify the points where the object connection will not work and suggest alternatives*
- identify where you can build different parts of the final object in parallel*

OO1 is the ideal environment for bringing about mass customization and enhanced productivity."

 <?php include '../HTIcopyright.php';?>