

Building Enterprise Information Management Systems using MANTRA – A Collaborative Development Environment

By:

Nitin Uchil

BDrive Inc., Santa Clara, USA

Summary

Enterprise Information Management consists of creating applications that can do:

- Files Management
- Content and Document Management
- Process Management
- Data Management

Herewith, we present MANTRA, an Integrated Development Platform that coordinates the different activities of Application Development (Plan, Define, Design, Build, Test, Deploy and Maintain). These applications are Knowledge Management Infrastructures provided to the Engineering community to reduce the inefficiencies in the Product Development Phase of PLM (typical to the Automotive and Aerospace vertical) and to optimize the Design process. MANTRA is web enabled, and manages all of the aspects of application development via a single interface including: code, reusable components, content, discussion forums, email, issues management, lessons learned and best practices, and enables a collaborated environment for enabling the application to be robust, scalable and sustainable.

Using MANTRA, we have created more than 30 applications encompassing Benchmarking, Test Data Management, CAE Simulation, Process Management and Prediction Utilities for different attributes of automotive engineering – Aerodynamics, Safety, Noise & Vibration. Climate Control, Durability, Engine Dynamometer Testing, Powertrain, Cooling & Heat Management, Vehicle Dynamics (Ride & Handling) and Packaging.

Keywords

KBE, IDE, UML, XP, MDA, Java, J2EE, XML, Expert Systems, RDBMS, PLM, PDM

1. Introduction

Providing Information Capabilities and Knowledge Management in Engineering requires:

- Engineering Domain Understanding
- Knowledge of the Business Process
- Ability to create Information Technology Infrastructures
- Capability to provide for Analytics
- Deployment of the solution to the Global community

1.1 The Engineering Domain

Engineering in the Automotive and Aerospace industries is the provision for vehicles to meet or exceed requirements set by government regulations, satisfaction of the customer, best-practice initiatives, inclusion of new technologies and the capability matrix of the vehicle. Engineering attributes like Safety, Aerodynamics, Powertrain, Heat Management, Vehicle Dynamics, and Noise and Vibration have over the years developed strong simulation capabilities that enable the analysis of functional requirements without the need for prototype build. A need to understand the Engineering domain is therefore an absolute must for applications that serve the automotive and aerospace vertical. This includes the physics and mathematics behind the simulations, understanding of the capabilities of the software packages and their limitations, how well they correspond to physical testing, and a list of capable vendors that can model and analyze the functional requirement.

1.2 Business Processes and Information Technology

Most of the Engineering activities are now outsourced by OEMS (Major Automotive Companies) to vendors specialized in these activities or ones that can provide cost effective off-shore capabilities for design, modeling and analysis of full vehicle/component characteristics. Engineers at the OEMs act as managers to facilitate the transfer of CAD data to the vendors, the assessment of the quality of the mesh and the cataloging of the reports of the analysis.

Also, with the constant demand to reduce cycle times in product development, there is need to manage information effectively and make it available to engineers on demand via a secure Portal that prevents illegal access and provides for just the right information to have analysts/planners make the correct decision(s) and sign-off on the requirements. To create IT infrastructures, companies have to be constantly aware of the protocols available (J2EE, .NET, Perl, other proprietary), their standards and best practices, and their capabilities and limitations.

Currently, developers are forced to use a combination of different tools to effectively build applications. These range from Project management tools like Microsoft's Project and Excel, tools from Rational Software (now part of IBM's software group) for issues management, application design and versioning, Integrated Development Environments like JBuilder and Database Schema builders like ERWin.

1.3 The Role of Analytics

Engineering analysis also deals with complex mathematics that comprise of reporting and post-processing of information. This has to be represented meaningfully and cascaded to upper level management so that the health of the processes can be ascertained and provisions for the future can be made. Analytics can be defined as an aggregation of On-line Analytical Processing (OLAP), Business Intelligence (BI) and Statistical Analysis with the provision of Unit Conversions, Charting and Spreadsheet type tabular output.

1.4 The Need for an Integrated Development Platform

What was needed is a tool that:

- Aggregates information from disparate sources,
- Enables management of the business process,
- Facilitates application build
- Has the capability to reuse predefined structures
- Supports Analytics and
- Orchestrates the different activities

via a single interface.

MANTRA, an Integrated Development Platform created by BDrive fills that need. Built on the solid software engineering construct of Organic Programming (OP), it effectively manages the Engineering of Software to build robust, flexible, scalable and sustainable applications. Depicted below (Fig. 1) is the process flow of assimilating information from different sources, aggregating them as a multidimensional cube, developing the application and representing it to the community.

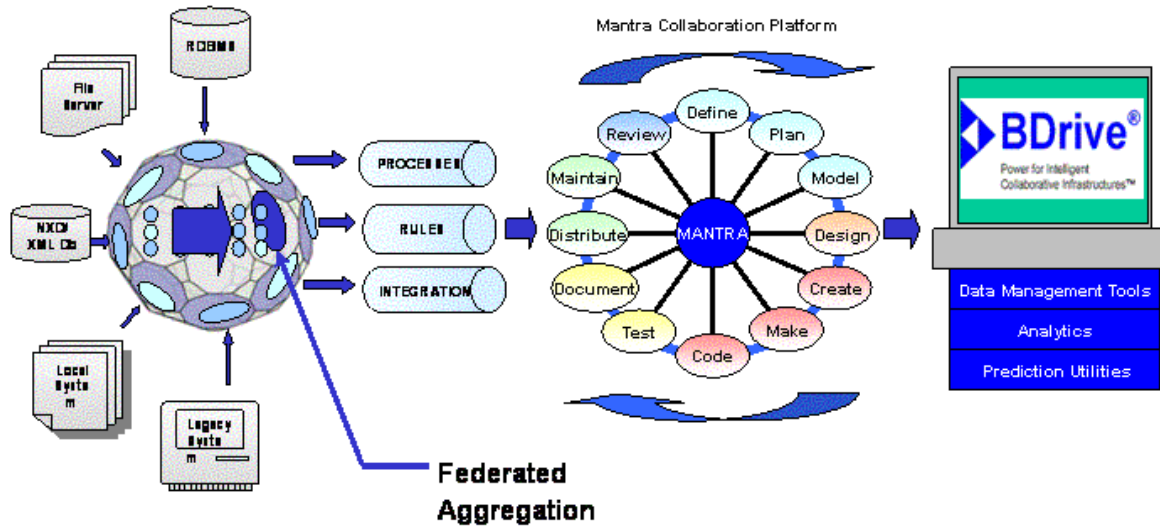


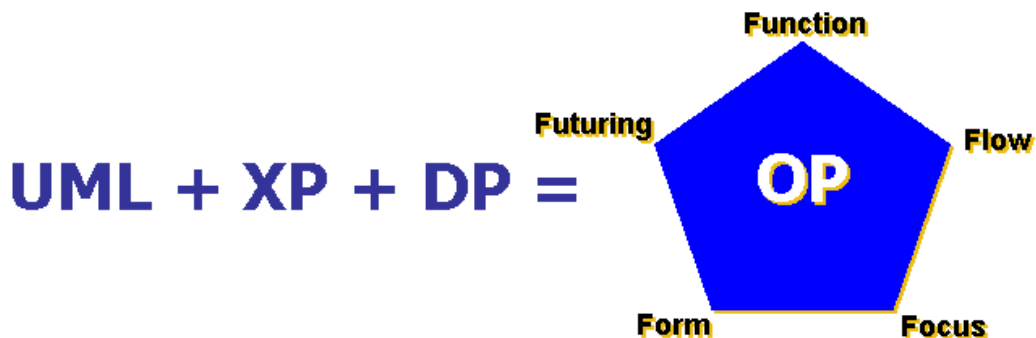
Fig 1: The MANTRA Integrated Development Platform

2. Organic Programming

2.1 What is Software Engineering?

Software Engineering is defined as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software. It deals with applying methods and scientific knowledge to create practical cost-effective solutions for the design, construction, operation and sustenance of software and associated products. Engineering is the methodology of applying math and science to synthesize a constrained solution to the needs of a customer. Software engineering is applying these principles to software.

2.2 The Definition of Organic Programming



“It is not a process of addition, in which performing parts are combined to create a whole, but a process of unfolding, like the evolution of an embryo”

- Christopher Alexander

Fig. 2: Organic Programming

Organic Programming (OP) is a new software methodology (defined by the author) designed for constructing complex yet flexible structures of program modules. It is about a new philosophy of software engineering - where form and function are one, where documentation and code are written in unison. Much like Organic Architecture, it is an attempt to integrate modules into a coherent whole: a marriage between the programming construct and its functionality and a union between the abstraction and its implementation. It is organized about the concepts of 5Fs (Function, Flow, Focus, Form and Futuring) to build robust, scalable and resilient software architectures. It reinforces the need that software development should be iterative rather than following the waterfall approach of current practices. Organic Programming is actually a deliberate and disciplined approach to software development. As shown above (Fig. 2) it combines the coding structures of the Unified Modeling Language (UML) and the best practices of Extreme Programming (XP) to bring about object-oriented code that is component based and bug free. It follows the philosophy of Gestalt - "The whole is greater than the sum of its parts". It stresses on a "Zero Administration Client" environment once the application is production deployed. It is an assimilation and re-organization of best practices followed by BDrive, Inc. in its projects of building knowledge management frameworks at major Automakers to optimize processes in the design and product development phase of Product Lifecycle Management (PLM).

Based on solid foundations of the Software Design Philosophy of OP, MANTRA uses tenets of *UML*, *XP*, *Agile Software* and Design Patterns (application of time-tested reusable components) development practices to transform rote code to automated procedures so that the users need concentrate only on the business logic when developing applications. OP incorporates the lifecycles of waterfall, staged, evolutionary delivery, incremental and spiral approaches to software development to create robust, scalable and flexible systems.

3. The Engineering of Software

3.1 Web Application Frameworks

Web-applications development involves a complex interaction of industry standard APIs, protocols, programming, markup and query languages, connection to disparate sources and content creation tools. Building even modestly complex web applications requires a team of developers in a variety of roles with a broad range of skill sets. Typically, this development complexity is managed with a class of software called "*web application frameworks*". However, current frameworks tend to focus on limited ranges of development roles/skills, or require specialized knowledge in order to be used productively.

What is really required is a web-applications development platform that facilitates process flow amongst the various stages in the development process, provides a basic framework for the application, allows for integration and collaborative structures to be established with external and legacy systems, reuses services and code, and most of all enables a orchestrated, coordinated and collaborative application build.

3.2 Progress Metrics Dashboards

MANTRA dynamically creates a progress metrics dashboard based on data collected from your application build. All team members benefit from having an easy-to-use and always-updated place to go for all project information. It catalogs all discussions pertaining to the project in one place for future reference. MANTRA also presents the results graphically so that the project's progress and quality are easily assessed. This allows for better prediction of which areas would require special attention and where to focus scarce resources to stay on schedule, and enables managers to make decisions based on quantitative analysis. Accurate and accessible project status information allows project leads to better anticipate problems or delays and make informed decisions in order to avoid them. Better project predictability helps to keep projects on schedule and on budget. The Manage entity also enables monitoring access to a project with its built in login feature. This built-in security prevents unauthorized access/ use of project information. It also enables project managers to determine who is working in what phase of the project and re-allocate work if appropriate. It also offers a web-based bug tracking system to report, assign, solve, test and close bugs.

3.3 Phases in MANTRA

Fig. 3 illustrates the phases in the Integrated Development Platform that enable the creation of robust IT infrastructures via a themed iterative process that accounts for Engineering Business Processes; Requirements Analysis; Visual Depiction of Components; Auto Generation of Code, its Testing and Build; Collaboration via facilities like File Upload/Download, Email, Issues Tracking and Discussion Forum; and the Management of the Work-plan. Included in the Framework of the Development Platform is the ability to define and create backend database entities; connection to these/other relational databases, PDM systems and other Legacy Systems and Code Templates that consist of reusable components that facilitate Query representations and Analytics.

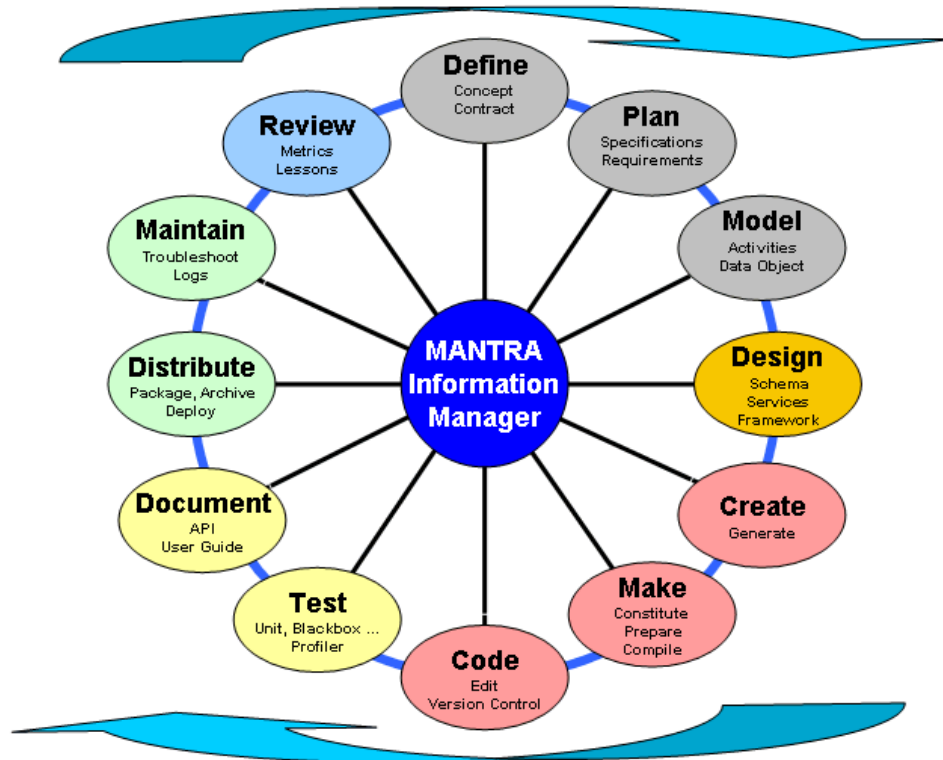


Fig. 3: The MANTRA Development Cycle

What MANTRA enables is the complete Engineering of the Software application and the cataloging of its components. The creation of the application is collaborative, issues related to IT standards are transparent to the developer and a fully cataloged list of time-tested design patterns are available on demand to be included in the purview of the application build. This enables the developers to concentrate on the Business Logic, integrate their individual code segments and concert the development of the application so that it is provided to the client in a timely and cost effective package.

4. Features and Benefits of using MANTRA

4.1 Web Based Integrated Development Platform

MANTRA provides a web-based solution that handles all aspects of software development from System Design, Planning, Development, Documentation, Testing, Distribution, Maintenance and Reviews. Being web based all project members can access the development platform from anywhere. It also provides for iterative development involving the client in every stage, ensuring that at all times development is resulting in deliverables, per client requirements. Changes can also be incorporated with ease, and implemented seamlessly throughout the development process.

4.2 Central Repository of Project Information

MANTRA provides a single place to store and view project information – articles related to project, project meta data, team members and their roles, milestones in project development, emails related to

project etc. Storing information in a central repository enables all project stakeholders to be abreast with latest developments in the project at any given point of time.

4.3 Web-based Collaborative Project Management

Being a web based development platform MANTRA provides for seamless collaboration between geographically dispersed team members. Members can view requirements submitted to them, share files, import and export code, compile and execute code, edit code etc. from virtually anywhere. MANTRA has in-built mechanisms for coordinating and integrating code from different team members, versioning etc.

4.4 Visual Modeling Interface

MANTRA provides a visual modeling interface enabling developers to visualize the architecture and relationships of an application.

4.5 Automated Code Generation

MANTRA auto generates a significant portion of the code resulting in accelerates development. Auto generation is enabled using MANTRA's proprietary design patterns technology. MANTRA also auto generates connectivity to external sources such as databases, file services etc. Design patterns also infuse the reuse of components optimizing the code development process. Developers can build libraries of components and select and customize them according to application requirements.

4.6 Monitoring Mechanism

MANTRA provides a web based monitoring mechanism to administer sources and web-servers. Users can view active sources and test them, start and stop web-servers with a web-enabled interface. The *Monitor* also provides ad-hoc query generating capability across multiple databases. MANTRA also automatically summarizes code by category, so that project stakeholders can gain an understanding of the application development effort. MANTRA provides for backup and storage of application data. MANTRA also has an in-built project manger, which manages the timeline and progress of the development effort.

4.7 Built-in Testing Capabilities

MANTRA has built-in test utilities that can perform different types of testing – source, unit, module, black-box, white-box, stress, regression and optimize. Testing is automated eliminating manual effort to do testing. Testing can also be implemented side-by-side with development, ensuring quality of the code.

4.8 Document Management

MANTRA handles all document needs of application development – User Guides, Javadoc API's, Reference Manuals, User Guides etc. using LIPI – BDrive's proprietary knowledge management technology. LIPI provides for document collaboration, web site content and versioning using a web based interface. LIPI not only to classify content related to application builds but also represents procedures, methods and encyclopedias. LIPI is essential to manage the content bits that influence the whys of an application build and to harmonize the best practices for future reuse.

4.9 Distribution, Maintenance and Reviews

Applications are deployed and maintained using a web-interface. Maintenance is facilitated through a request management system that is coupled to an internal Bug-tracking Utility. Clients can submit issues and monitor the progress of the issue using a web-enabled interface. MANTRA also facilitates cataloging of best practices and methodologies, lessons learnt etc., so that knowledge gained through each development effort is not lost.

4.10 Key Benefits of Using MANTRA

- One integrated platform for the entire development process.
- Web-enabled collaborative development process, which can unify geographically, distributed development teams.
- Automates most of the development process resulting in significant savings in time and cost.
- Providing expert-level insights with guidance on their proper application and adaptation using pattern technology
- Iterative and highly flexible client-driven development
- Significant reuse of components throughout the development process

- Streamlines development by integrating with leading IDE's
- Development becomes highly collaborative and all stakeholders can gain deep visibility into the progress of the project at any given point of time.
- Complete open standards approach integrating with all operating systems, existing legacy systems and relational databases

5. Test Data Analysis - An Example Application Created Using MANTRA

Testing of competitive vehicles, production cars and prototypes by full vehicle, system, subsystem and component is the most expensive facet in the product development lifecycle of automobiles. Storing and retrieving test information, if optimized can provide enterprises with the competitive edge they need to be leaders in their domain. This would also provide the ROI on the expenditure in doing expensive tests. This section discusses the tenets used in uploading test data and representing it using web enabled, database driven and security governed utilities.

5.1 The Process

Testing involves management of meta-data and the test values that stream out of the test bed. Meta data may consist of information regarding the test or vehicle, or the set-up used to run the test. A correct comparison of different tests can only be facilitated if the conditions are similar. Due to the large amounts of data and the total number of tests conducted, it was necessary to build a web-front end that consisted of:

1. Cleaning up raw data and summarizing it
2. Uploading the final data structure to a back-end Oracle database. This included data values, reports and media files related to the test
3. Administering the information and making it available to groups/individuals based on a security model
4. Viewing the information in easy to use templates with metadata discovery, graphing and analytics.
5. Retrieval of the Information for downloading to Excel for further manipulations.

Shown below in Fig. 4 is a typical Test Data Scenario (used to manage Noise and Vibration Information of Vehicles Tested).

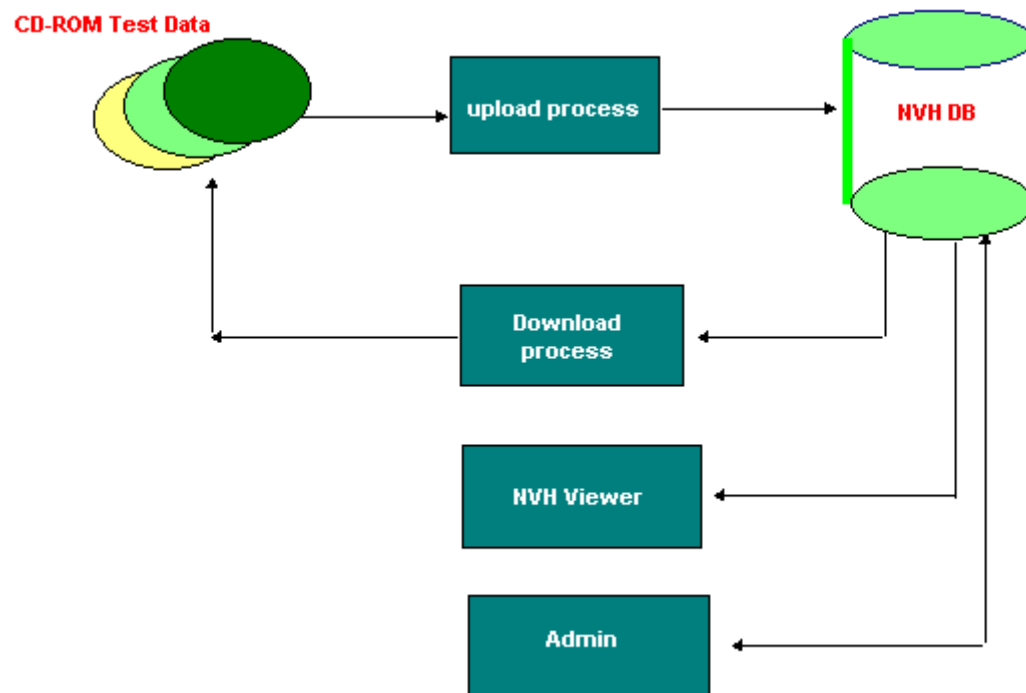


Fig. 4: Attributes of a Typical Test Data Analysis

5.2 MANTRA Elements

MANTRA Integrated Development Platform 4.0.0 PROJECT: NVH Log Out NUCHIL

Project-NVH

- Information
 - Meta Data
 - Team
 - History
 - XML File
 - Home Page
 - Utility
- Manage
- Define
- Plan
- Model
- Design
- Create
- Make
- Edit Code
- Test
- Document
- Distribute
- Maintain
- Review

Show Team By Phase

Phase	Authors	Designation
Project	CSOUNDAP, NUCHIL, PKRISHNA, RPHADNIS, SGOPAL	Team Leaders
Project		Clients
--Manage		Administrators
--Define		Planners
--Plan	CSOUNDAP	Planners
--Model		Modelers
--Design	CSOUNDAP, NUCHIL, PKRISHNA	Architects
----Schema	NUCHIL, PKRISHNA, SGOPAL	ERD Specialists
----Services	NUCHIL, RPHADNIS, SGOPAL	Connection Specialists
----Imports		
----Map		Integrator
----Persistence		Integrator
----Business Logic		Component Specialists
-----POJO		
-----EJB		
-----Framework	NUCHIL, PKRISHNA, RPHADNIS, SGOPAL	Component Specialists

Fig. 5: Enabling access for Development via Team Definition

MANTRA Integrated Development Platform 4.0.0 PROJECT: NVH Log Out NUCHIL

Project-NVH

- Information
- Manage
 - Meetings
 - Emails
 - Discussions
 - Shared Drive
 - Bug Track
 - Access Log
- Define
- Plan
- Model
- Design
- Create
- Make
- Edit Code
- Test
- Document
- Distribute
- Maintain
- Review

After-Life Metrics ? Apply

Metrics

ITEM	VALUE	ACTION
Number of Projects	1	
Number of Authors (Developers)	6	
Begin Date	02/27/2002	
Total Number of Issues	144	
Last Activity Date	06/02/2003	
Average Life (Closed - Discovered)	12.99 days	
Average Wait (Assigned - Discovered)	6.51 days	
Average Passive (Start - Assigned)	4.69 days	
Average Active (Resolved - Start)	-01 days	
Average Dormant (QA Phase: Closed - Resolved)	1.79 days	
Total Fix (Authored) Time	268.05 hours	
Average Fix (Authored) Time	1.86 hours	

Date(MM/DD/YYYY) Type Phase Author Project

From: To: All All All nvh

Life Span >: Keyword Search: Report: Metrics **Apply Filter**

Fig. 6: Issues Management

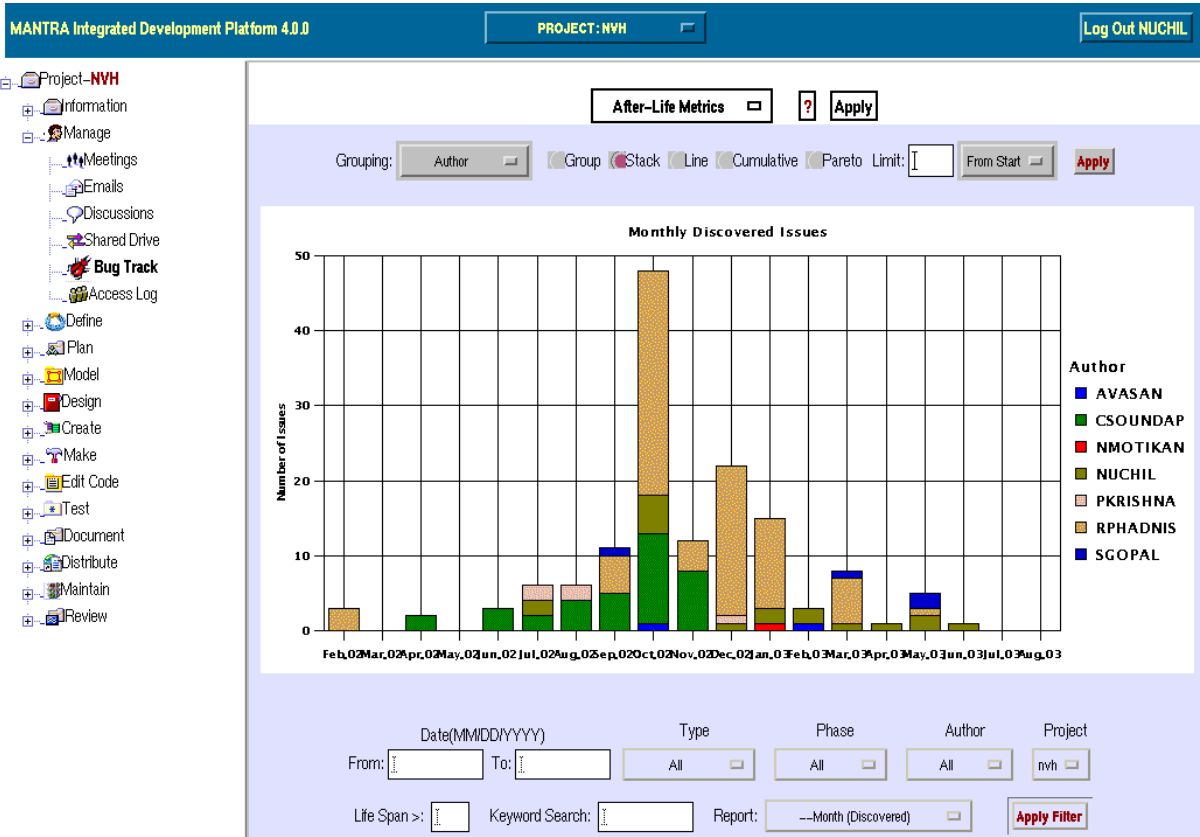


Fig. 7: Summary of Issues by Month/Developer

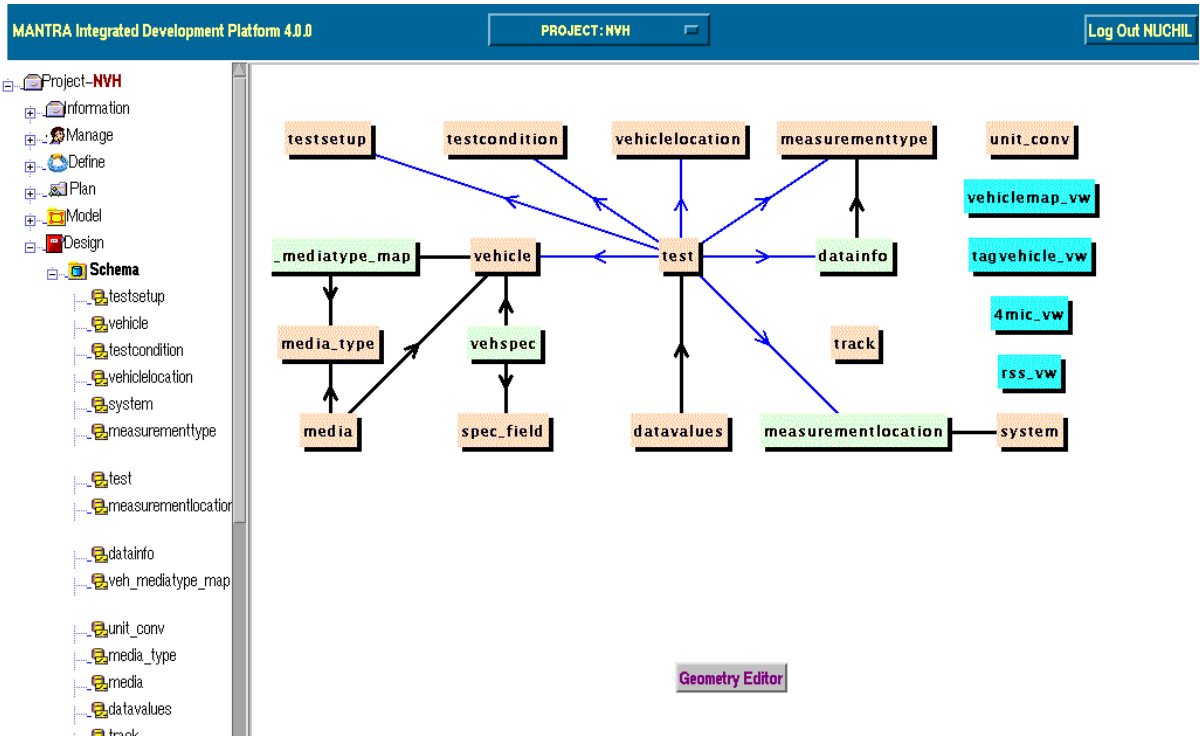


Fig. 8: Schema (Database) Definition

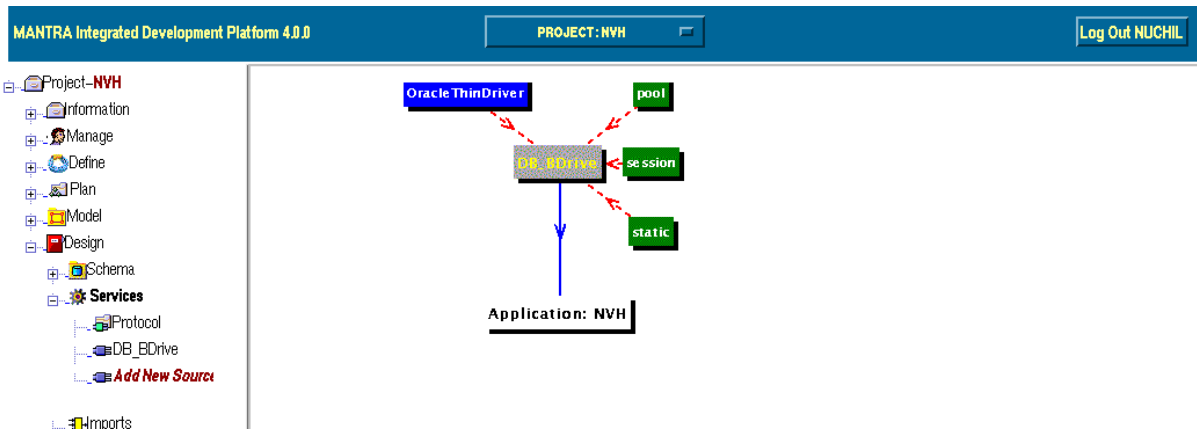


Fig. 9: Service Definition (Source Connection Properties)

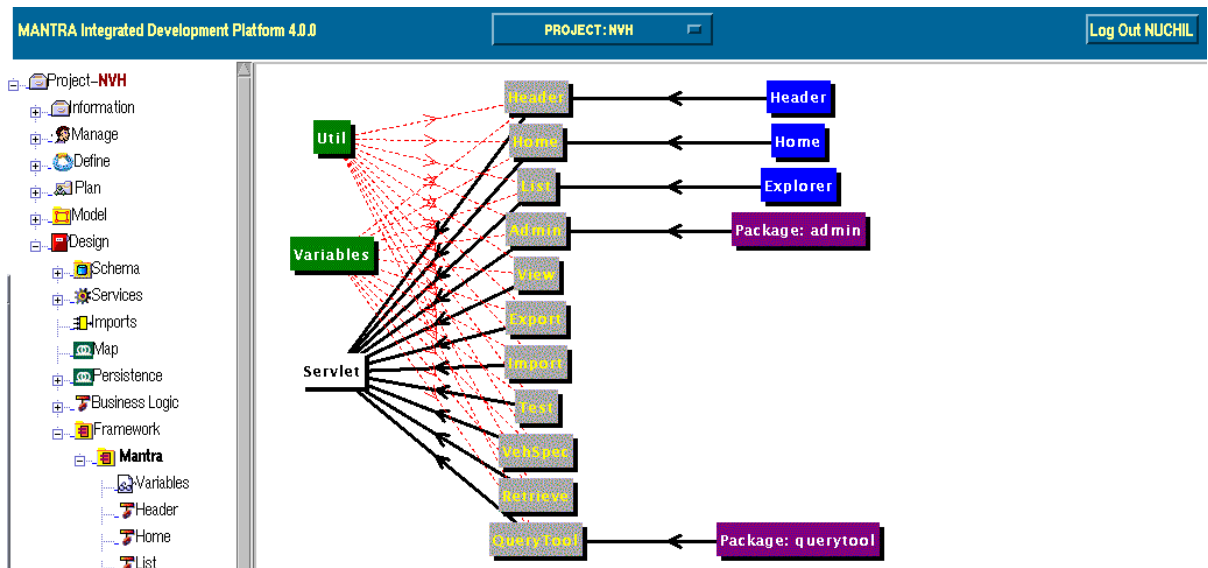


Fig. 10: Components of the Application Framework

5.3 Application Snapshots

Welcome to the Powertrain NVH Database

Please Maximize your browser window when using this application.
Also, close any menu bars that you do not need from the browser window

Please Choose An Item From The Pick List Above

Import: Copy processed vehicle/powertrain directory structures to the server for upload.

List & Load:

- List the directory structure and navigate through it.
- View the files and load data to database.

View: Select by Test or Vehicle/Powertrain and view as either Plot, Table or Excel/CSV file.

- Test centric View:** Select test attributes and then vehicles/powertrains.
- Vehicle centric View:** Select vehicles/powertrains and then test attributes.
- Noise & Vibration View:** Select vehicles/powertrains and then relevant noise and vibration channels.

Retrieve: Select tests to be retrieved from the database as CSV files to the Export file location.

Export: Select CSV files in the Export file location to be downloaded.

Fig. 11: Home Page of Application

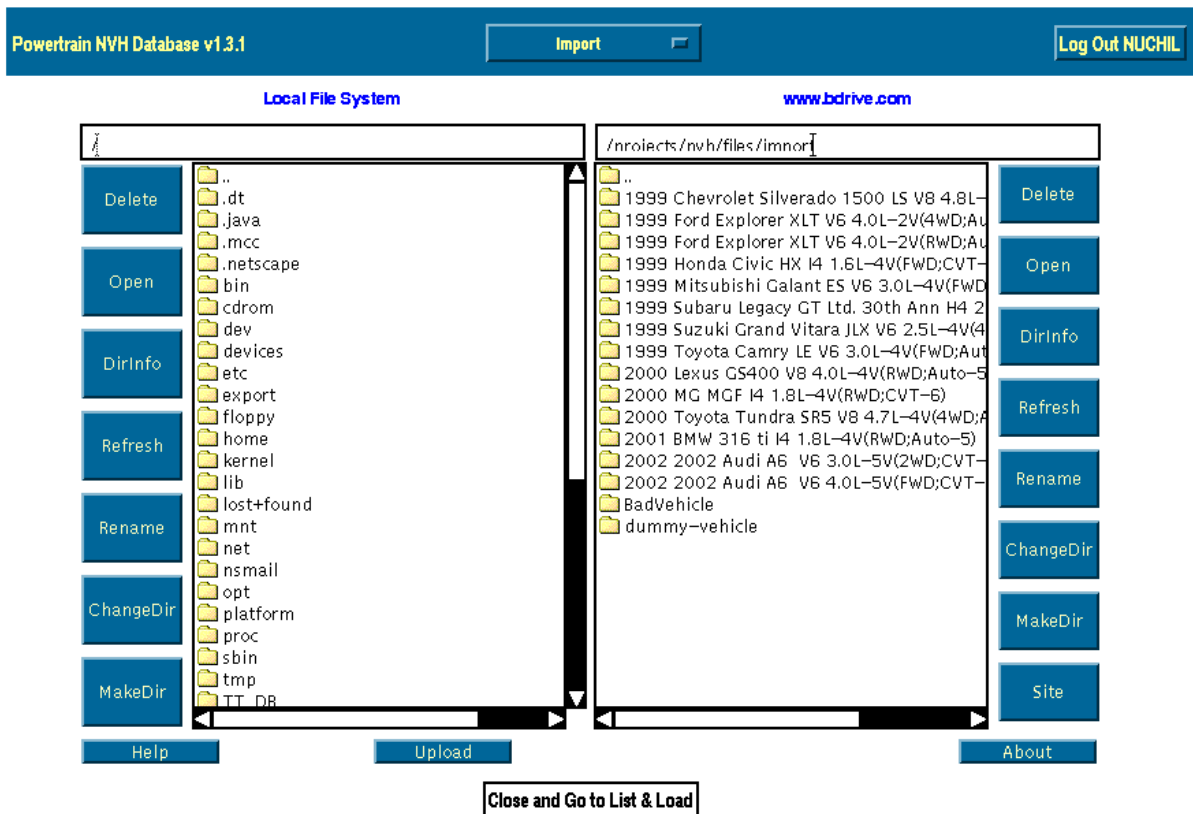


Fig. 12: FTP Upload of Vehicle Test Directory Structure

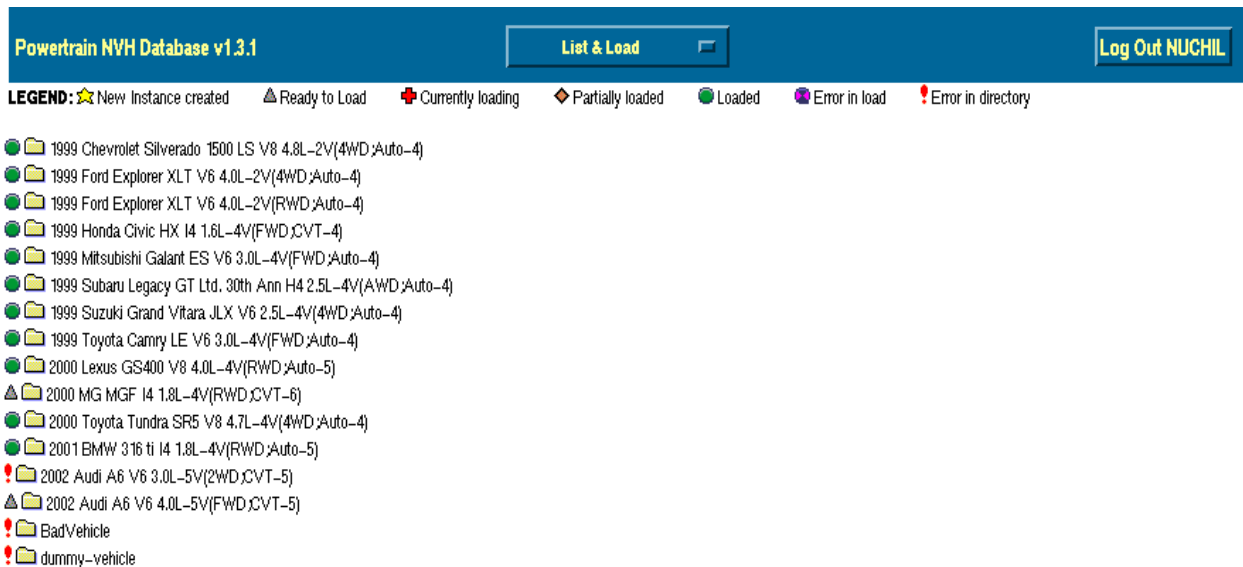


Fig. 13: Listing of Vehicles with Load Information

Powertrain NVH Database v1.3.1 _ Vehicle centric View [Log Out NUCHIL](#)

<p>1999 Mitsubishi Galant ES V6 3L-4V(FWD;Automatic-4)</p> <p>1999 Subaru Legacy GT Ltd. 30th Ann H4 2.5L-4V(AWD;Automatic-4)</p> <p>1999 Suzuki Grand Vitara JLX V6 2.5L-4V(4WD;Automatic-4)</p> <p>1999 Toyota Camry LE V6 3L-4V(FWD;Automatic-4)</p>	<p>Logic</p> <p>OR <input type="checkbox"/></p> <p>Test Setup</p> <p>Vehicle(Chassis Dyno) <input type="checkbox"/></p> <p>Test Condition</p> <p>WOT_1st Gear <input type="checkbox"/></p>	<p><input type="checkbox"/> Use Point ID for Measurement Location</p> <p>Vehicle Location</p> <p>Engine Compartment <input type="checkbox"/></p> <p>Measurement Type</p> <p>Noise <input type="checkbox"/></p> <p>System</p> <p>Engine <input type="checkbox"/></p> <p>Measurement Location</p> <p>FP Engine Mic_Bottom_Local_Softm(EC), Under Hood_Dash_Center(EC)</p> <p>Data Information</p> <p>1/3 Octaves(51.5Hz-100Hz)(dB), Overall Levels and Order(0-5-20th)(dB), Overall Levels(dB)</p> <p>Test Options: NULL <input type="checkbox"/></p> <p>Apply Selections</p> <p>Extract Channels</p>
---	---	---

<p>X Axis</p> <p>Engine speed (rpm)</p>	<p>Y Axes</p> <p>OA_Level</p> <p>dB(A)</p> <p>dB(C)</p> <p>Pa</p>	<p>Vehicle/Powertrain Info Levels of Display</p> <p><input type="checkbox"/> Level 1 <input type="checkbox"/> Level 2 <input type="checkbox"/> Level 3</p> <p><input type="checkbox"/> Year <input type="checkbox"/> Make <input type="checkbox"/> Model</p> <p><input type="checkbox"/> VIN <input type="checkbox"/> EngineID <input type="checkbox"/> TransID <input type="checkbox"/> Mileage</p>	<p>Review Vehicle/Powertrain</p> <p>Plot Data</p> <p>View Table</p> <p>Export to CSV</p> <p>Export to Excel</p> <p>Show Test Setup</p>
--	--	---	---

Fig. 14: Selection of Test Data using the “Vehicle” Criterion for View

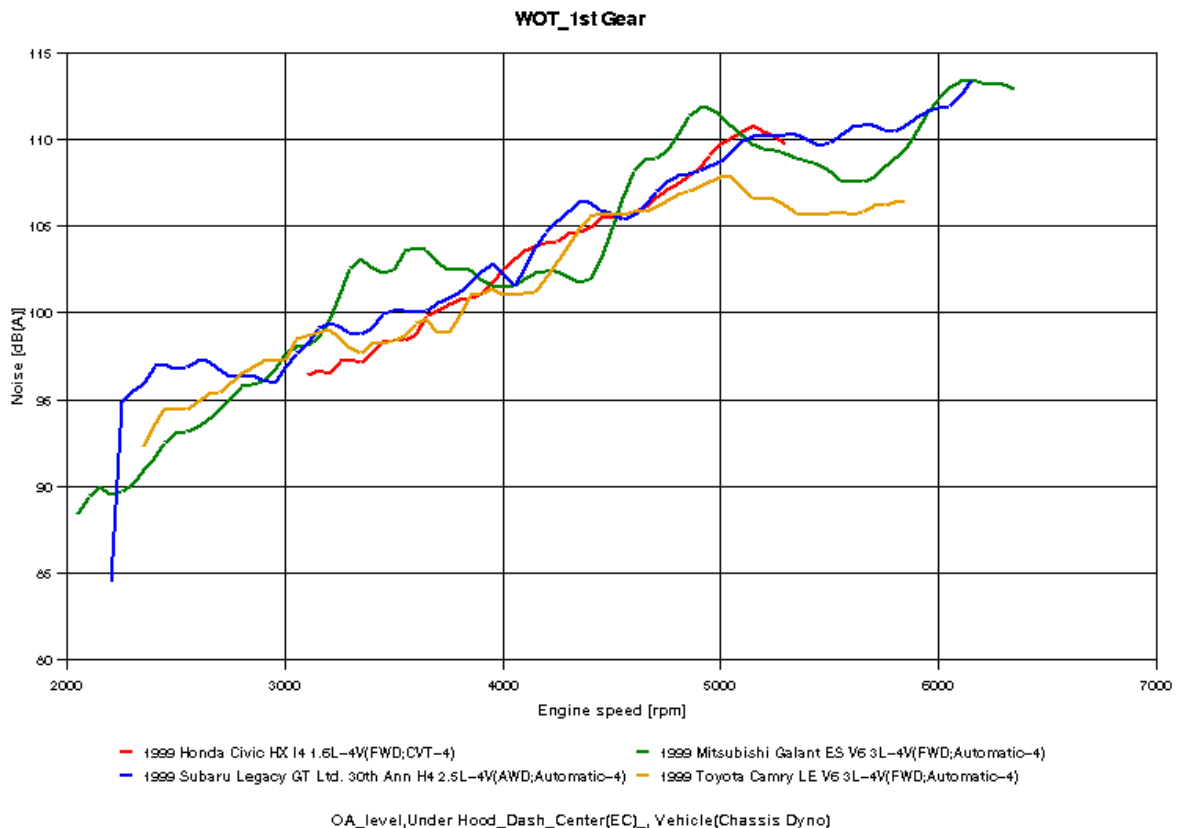


Fig. 15: Trace Plot of Noise Characteristic for selected vehicles

6. Conclusion

In today's knowledge-based economy, success depends on sharing relevant, timely, and complex information with employees, customers, and partners. Knowledge-based Internet solutions are the only way businesses will exist in the world of tomorrow: be it serving clients, transacting businesses, enabling employee communications, supply-chain management or manufacturing products.

Extending the enterprise, enhancing business relationships and sustaining a competitive edge, all depend upon a flexible, scalable, reliable e-Business strategy, backed by a sound infrastructure. Transforming the organization of a "bricks-and-mortar" business to fully take advantage of the dynamic Internet business economy is a challenging task - but one that is vital to any business longevity.

MANTRA would provide for building KBE infrastructures and benefit medium-to-large organizations by:

1. *Improving speed of IT execution through:*
 - Simplified and centralized evaluation, acquisition, deployment, management and support of multiple middleware resources
 - Fast deployment of services without disruption to business
 - Ability to quickly and securely add or modify IT capabilities, users and locations
2. *Improving focus of IT resources through:*
 - Simplified IT operations and management, so that IT departments can spend less time in maintenance and "firefighting" modes and more time focused on strategic IT initiatives
 - Better IT performance, increasing operational efficiency and end-user productivity
 - Improving IT cost management through:
 - Streamlined operations through automation of rote IT tasks and business processes ,thereby, reducing operational costs
 - Proactive 24x7 system monitoring to prevent costly problems such as server overloads or crashes from occurring
 - Reduced initial cash outlays for new technologies by leveraging existing IT resources and accessing technology as an operating expense instead of as a capital expense
 - Scheduling of predictable fees on a periodic basis in a single bill
3. *Improving end-user satisfaction through:*
 - Service-level commitments backed by 24x7 support and continuous performance monitoring activities, providing uninterrupted service to users
 - Customized and personalized portal access to IT services from virtually anywhere in the world.
 - Use of broadcast features to notify users of impending timelines or administrators of service interruptions.
 - Continuity of applications management through robust and sustainable methodologies

MANTRA provides an ideal environment for developers to create applications in a rapid development framework. Consultants can use it to architect their solutions as they are creating strategies and process maps for enterprises. Since the platform is web enabled, collaborative development and management of the application are facilitated so that large-scale infrastructures can be built.

7. Contact

For further information, please contact Nitin Uchil at nuchil@bdrive.com.