TIES: An Engineering Design Methodology and System

Lakshmi S. Vora, Robert E. Veres, Philip C. Jackson, and Philip Klahr

The technical information engineering system (TIES) is a computer-assisted methodology to help achieve significant improvements in product quality, production cycle time, and market share. TIES builds on the foundation of quality function deployment (QFD), a systematic method of ensuring that customer demands are accurately translated into appropriate technical requirements and actions. TIES is a design tool and provides a framework in which cross-functional product-process design teams can collect and store relevant engineering information, experience, and knowledge. This common framework allows separate groups to design components or elements while they maintain consistency with the designs of other groups. TIES incorporates several AI techniques, particularly in the area of knowledge acquisition and knowledge-based system development. TIES represents one of the first deployed systems that is both a knowledge-acquisition tool focused on engineering design and a system to support team design. TIES is being used to support the design applications varying from specific automobile components (for example, an instrument panel) to a complete new vehicle.

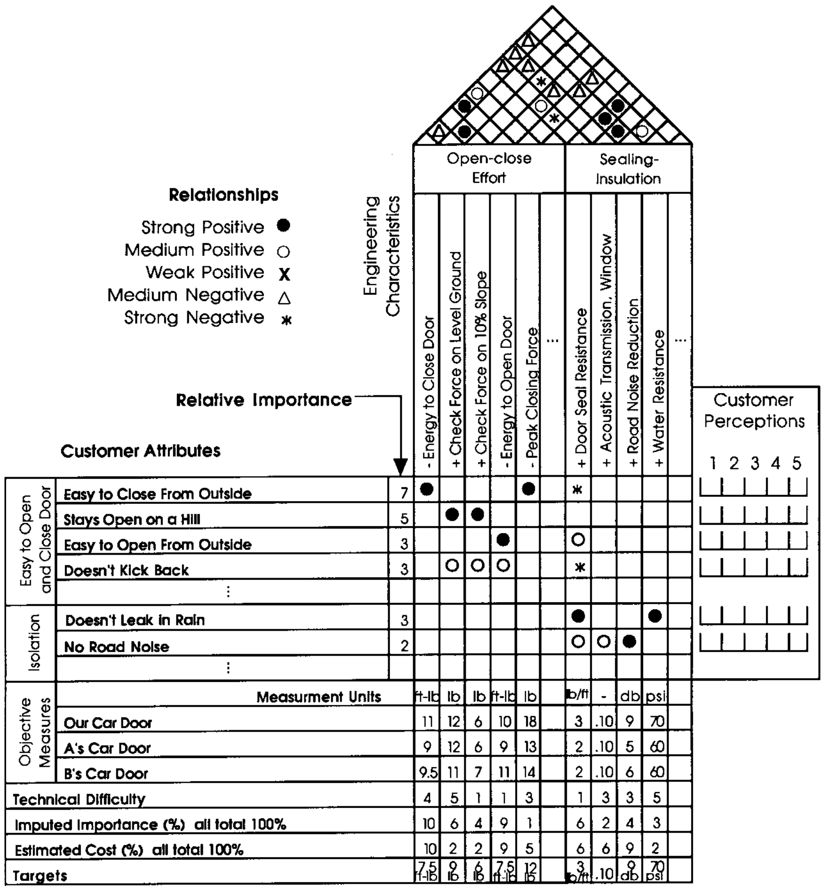

Figure 1. Quality Function Deployment House of Quality.

Background

Product quality and design cycle time are two critical factors in achieving customer satisfaction and market share. Nowhere are these elements more important than in the automotive industry. QFD, a structured technique developed in Japan (Hauser and Clausing 1988; Sullivan 1988), provides a systematic means of ensuring that customer desires are accurately translated into appropriate technical requirements and decisions throughout each stage of product development. QFD provides the basis for a planning tool and a process methodology. It helps to identify the most important product characteristics, necessary control issues, and best tools and techniques to use. When used

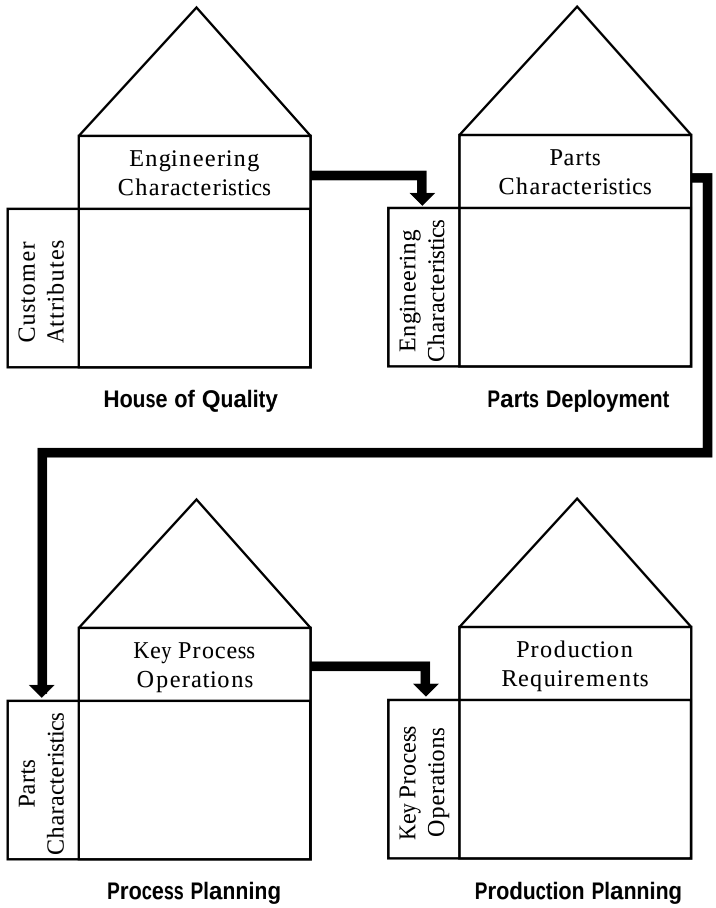

Figure 2. Cascading Houses in Quality Function Deployment.

throughout product development, QFD provides a comprehensive tracking tool and communications medium. Japanese companies credit QFD with improvements in both product quality and design time, resulting in increased market share (King 1987).

The working framework of QFD is a chart called the house of quality (figure 1). Matrixes display interrelationships between customer wants and technical know-how. These matrixes also provide information about weighing factors, comparative data, and correlations.

Fundamental to the house of quality is the assumption that products should be designed and developed to reflect customer tastes and desires. Marketing people, design engineers, and manufacturing personnel must all work closely, from initial ideas to product delivery. In QFD,

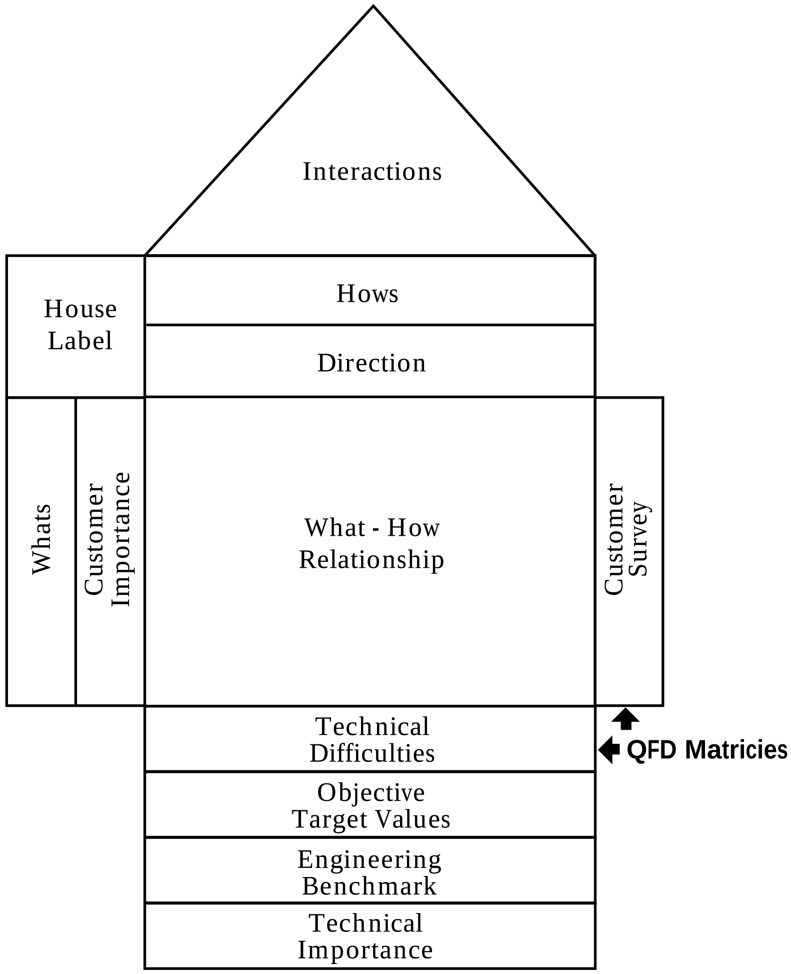

Figure 3. TIES House.

this process is facilitated by cascading through chains of related houses (figure 2) to link customer requirements to engineering characteristics and then to manufacturing processes. In this way, people with different responsibilities and problems can explore alternative designs and priorities while they maintain conceptual links to each other.

Overview of Ties

Ford Motor Company developed a computer-assisted methodology that automates and extends the QFD framework. TIES collects and stores relevant engineering information and human experience from cross-functional product and process design teams. It uses a knowledge-based system approach to facilitate design decisions, help resolve cross-functional issues, and retain valuable corporate engineering knowledge.

TIES provides a highly interactive development environment through a customized graphics system that allows users to create visual representations of houses that can be modified as often as desired. TIES enables users to define, refine, modify, and view cascades of houses that can have virtually unlimited scale (numbers of rows and columns and number of levels in the cascade). From a user's perspective, TIES offers numerous functions and capabilities, which are outlined in the following subsections.

User Development Process

A TIES house consists of a number of factors, including whats, hows, customer importance, technical difficulty, estimated costs, and objective measures (figure 3). Houses in TIES record the relationships between *whats* (customer or design concerns) and *hows* (necessary elements needed to address these concerns either through design or components), the engineering trade-offs, and the competitive position with respect to customer desires and objective measures. TIES supports the structuring of hows and whats into levels of detail and the automatic cascading of houses.

Aggregation and Sharing of Technical Expertise

Previously defined houses or their components can be used to generate new ones or can be merged to form a house with all the elements of the originals. This merging process allows partitioning a knowledge base among a set of experts and merging results to form complete designs. Also, information from an existing knowledge base (generic or specific) can be adapted to a new one.

Viewing and Analysis

TIES provides an interactive graphic facility for displaying and updating houses. (A hard copy can also be obtained for any graphic display.) To facilitate the viewing and analysis of houses, TIES provides a variety of focusing and filtering options as well as the ability to record individual user preferences for these operations. The following display options allow the user to configure the most relevant house components to be shown in the limited viewing screen:

Zoom lets the user decide, on an individual row and column basis, what level of detail is presented on the screen. *Filter* allows the user to

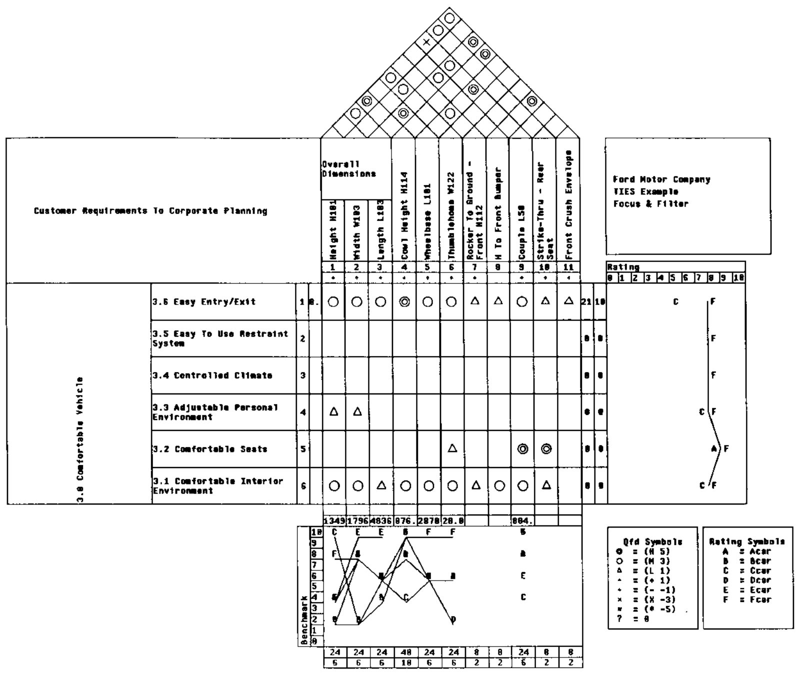

Figure 4. Example of TIES *Focus Feature.*

decide whether individual rows or columns should be displayed in the chart. *Focus* allows the user to select a specific concern and construct a subset chart that includes only the direct or indirect impacts for this concern. To illustrate, figure 4 shows a TIES house that originally contained 46 what items and 33 how items. With a combination of filter and focus operations, this number was reduced to 6 what and 11 how items, allowing much easier and more focused analysis.

The Exploration of Alternative Designs

Designers can readily explore alternative designs in several ways: Houses can be modified and impacts analyzed within a particular context, new contexts can be sprouted using copy and edit, contexts can represent a partitioning of the design and alternative partial designs merged as desired, and so on.

User-Supplied Annotations

Users can supply and examine explanatory text to present or find

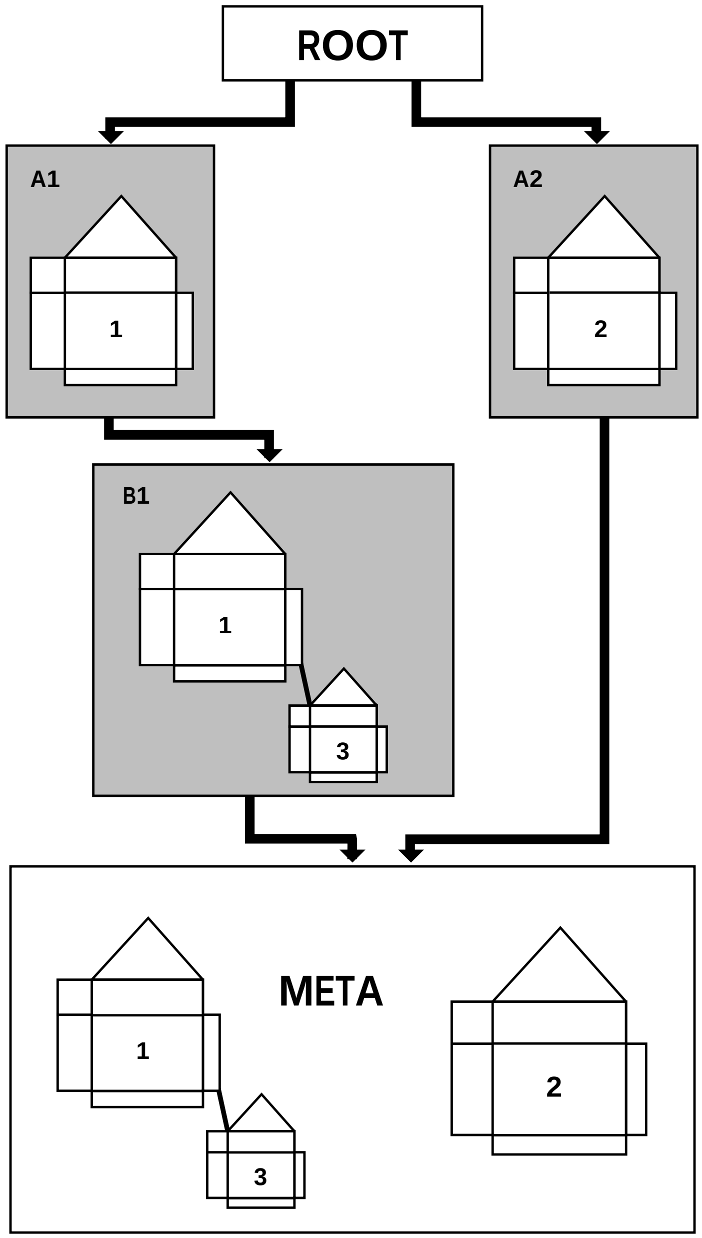

Figure 5. TIES Contexts.

more information about the whats, hows, and relationships. The source of the information (that is, user who entered it) is automatically logged by the system. Users can also attach information about relevant analysis and test procedures to support the data as well as create a "yellow pages" to specify who the experts are for the various house entities.

The Propagation of Qualitative and Quantitative Information

After a user enters information into a chart, user-defined rules can infer results and automatically assert other information into the chart. One can view this capability as a symbolic generalization of the numeric propagation technique found in spreadsheet programs.

TIES Architecture

TIES is deployed on Sun 4 workstations. Each Ford design engineer using TIES develops a knowledge base that is stored in a database. Entire knowledge bases or their components can be shared by design engineers or design teams. No other processes or databases are currently interfaced with TIES.

Contexts

All information in TIES is stored within *contexts* (figure 5). Each context contains a house or a chain of houses. TIES users can create, modify, copy, and delete contexts as needed. Contexts exist in a directed acyclic graph, which relates contexts with one another. Contexts inherit information from their parents (through multiple inheritance). Starting from a root context, which is always present, users can choose to partition their houses as desired, for example, by domain, project, or alternatives. In addition, TIES provides a persistent *metacontext,* which inherits information from all the contexts, as though it were the virtual child to all of them. The metacontext provides a global view of all contexts.

Frame Kernel

Contexts are represented as frames. TIES contains a customized frame kernel that automatically creates or deletes frames as users define, sprout, or merge contexts. The frame kernel incorporates a data-dependency logic (Charniak et al. 1987) to support multiple inheritance. Inherited data are tagged with support from the parent from which the data were inherited. Thus, inherited data are deleted from a context only when their support from all parent contexts is removed or when they are explicitly deleted from this context.

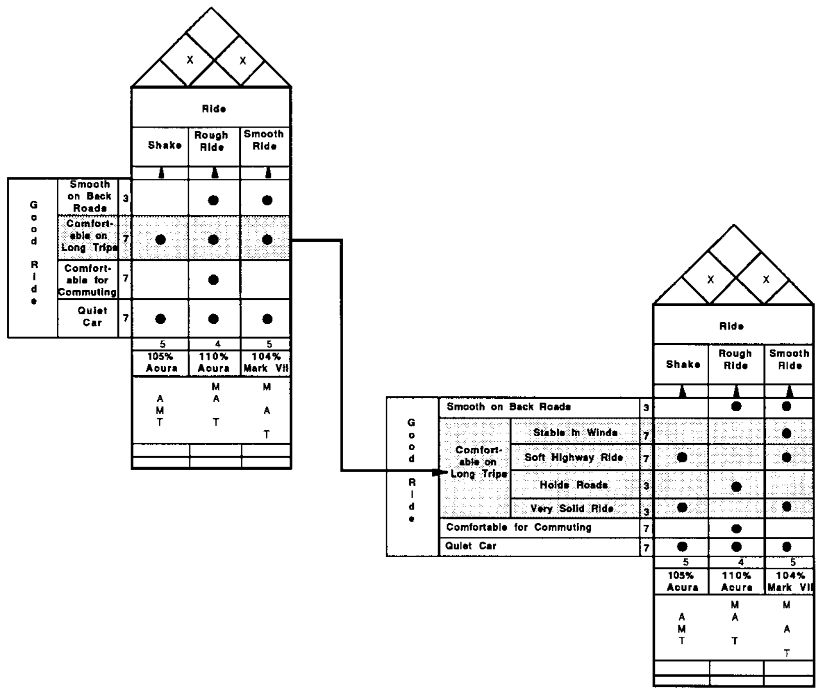

Figure 6. Conceptual Zoom.

The Representation of Houses

The structural information in a context defines a collection of data structures called entities, relations, and matrixes that together make up the one or more houses contained in the context. *Entities* are user-defined tree structures (of unlimited depth) that comprise the rows and columns of matrixes and houses. User-defined labels are text strings that are attached to entities for display. *Relations* are dynamic (user-changeable) mappings from pairs of entity nodes (row and column) to values. *Matrixes* are defined by entities of rows and columns and a TIES relation for storing cell values corresponding to pairs of entity nodes. Relations are implemented using extensible hash tables (Steele 1984). Thus, TIES matrixes can be extended automatically and transparently to the user. Finally, *houses* are a collection of related matrixes in a given context.

Hierarchical Abstraction and Conceptual Zoom

Row and column entities are hierarchically structured, with more abstract entities spawning more specific entities (as children). Thus, TIES enables users to perform various QFD functions at multiple levels of abstraction through a feature called *conceptual zoom* (figure 6). Tree structures can be truncated at any level of detail and for any combination of nodes. In addition, TIES permits the inheritance of cell values between matrixes viewed at different levels of conceptual zoom.

The Linking of Entities to Understand Impacts

TIES provides an analysis of which elements of a house would be impacted by changes to other elements. Two entities *impact* one another if a relation exists between them (direct impact) or if there is a chain of relations between them (indirect impact). Given a user-specified entity node, TIES searches for entity nodes in the context that are directly or indirectly impacted. The search can be constrained to a single house or a cascade of houses. Thresholds can also be specified to filter impacts. Results can be ordered, for example, by strength of the relationship to the initial entity.

Symbolic Inference Rules

TIES provides a customized, limited rule-based facility for users to define symbolic inference rules to propagate information (qualitative and quantitative) across matrixes. Note that impacts determine existing causal linkages, but rules actually propagate values.

Several AI representational techniques are incorporated within the TIES architecture. Frames (objects) are used to represent contexts that are organized in hierarchies (with multiple inheritance). Data dependencies are explicitly represented to maintain consistency within the context hierarchies. Knowledge hierarchies also exist for row and column entities to provide the conceptual zooming facility. Finally, a limited rule-based capability allows users to define constraints across matrixes.

Innovations

TIES provides innovations along a number of different dimensions, that is, from the perspectives of the design engineer, the design team, and the QFD application framework.:

From the viewpoint of the design engineer, TIES provides (1) a knowledge-acquisition tool customized for design, where the designer directly inputs his(her) own knowledge, thus avoiding the knowledge-acquisition bottleneck (in other words, no knowledge engineer is required); (2) user-defined knowledge representation, where the designer decides how to represent his(her) own knowledge within the frameworks provided, for example, particular entity structures, relations, and contexts; and (3) a symbolic spreadsheet capability, where designers can define symbolic inference rules to propagate qualitative and quantitative information within matrixes and across matrixes.

From the viewpoint of the design team, TIES provides (1) a common language and knowledge structure for all designers and (2) the decomposition of the design task and the merging of partial designs. Designers can separately work on components or various desired characteristics and combine their designs through merging. This approach is particularly desirable where design expertise is scattered among many experts.

From the viewpoint of the QFD application framework, TIES provides (1) one of the first deployed automated QFD systems (other companies might also be in the process of automating QFD features, but TIES is certainly one of the first to be deployed and is the only known QFD-based system that embodies AI programming techniques and knowledge-based system technology) and (2) significant new features to the QFD framework, including conceptual zoom, contexts, direct and indirect impacts, symbolic rules, and the generation of virtual QFD houses (produced by chaining cascades of houses).

Criteria for Successful Deployment

The single most important factor for the success of TIES was that it was driven by the needs of the user community. From the outset, the project had enthusiastic support from users; it continues to evolve as the number of users expands, and they provide feedback on modifications and extensions.

From a technical perspective, TIES is largely successful because of the flexibility it provides to users. In building knowledge bases, users can define their own structures on their own using their own terminology. Graphically, they can focus on pertinent issues or globally view total designs.

In terms of ongoing success, TIES will be measured based on the steady increase in the numbers of installations and users and in the growth and evolution of design applications. During the alpha and beta testing phases of TIES, approximately 50 to 70 engineers within Ford (United States and Europe) used TIES to support design applications varying from specific vehicle components, for example, an instrument panel, to a complete new vehicle.

1987	
January–May	Feasibility study for design process improvements
June	Ford proposal to use computer-aided QFD as design methodology
September–December	Project definition revisions; corporate review and endorsement
1988	
January	TIES project initiation; selection of Inference as technical partner
January–March	Proof-of-concept prototype (houses, matrixes, contexts) on Symbolics
April	Extensive demonstrations; request for system deployment by users
May–June	Enhancements (user interface, added features); first knowledge base
July–October	Enhancements; port to Sun; quality assurance testing
October	Sun version operational; limited-use release
October–December	Field test by vehicle program group
1989	
January	Alpha version operational; release to four installations
January–May	Continuing enhancements based on feedback from 15–20 users
June	Beta version operational
June–August	Completion of production version of Ties
August	Initial production version operational
December	Final production version operational

Figure 7. Development and Deployment Timeline.

Payoff

The entire life cycle of designing and producing a new automobile is several years. From this perspective, TIES is in its infancy. As new automobiles and their components are produced from TIES-assisted designs, we anticipate reductions in life-cycle times as well as improvements in product quality. A difficulty in estimating payoff is in determining the extent to which TIES-assisted designs prevent problematic designs from occurring in the first place. Actual measurements can only be done after years of experience, analyzing costs in correcting faulty designs and determining how these designs were developed. Again, we anticipate significant reductions in such costs because TIES provides a framework for cross-functional teams to design simultaneously.

Overall, expected TIES benefits include documenting, preserving, and internally distributing expert knowledge about design choices, engineering trade-offs, and customer concerns; providing consistency and coordination of vehicle program development; reducing development time for new vehicle programs; and accelerating the development of expertise to newly hired engineers.

Development and Deployment Timetable

TIES was designed and implemented by a joint Ford Motor Company and Inference Corporation team using the automated reasoning tool (Art) and Common Lisp. TIES is deployed on Sun 4 workstations. The development and deployment timeline for TIES is shown in figure 7.

The TIES project was carried out by a small team comprised of personnel from Ford's research staff, key Ford engineering users, and Inference Corporation working in a highly interactive manner to simultaneously develop TIES concepts, prototype software, and applications. Project management and coordination of TIES applications was performed by Ford's research staff. Software development personnel were added to the team to support quality assurance testing of the alpha and beta versions and to complete the production version, including documentation.

Time and costs for involvement of the Ford user community to help design and test TIES are difficult to estimate. As far back as the initial proof-of-concept prototype, users involved in the TIES project began building knowledge bases and designs that immediately enhanced their own operational efficiency and effectiveness. Consequently, from a cost perspective, user involvement costs were probably zero (or even negative) given the immediate productivity savings.

Summary

The TIES project combined engineering methodology, product design, and AI techniques to produce a new tool to support automobile designs at Ford Motor Company. In its current state, TIES has already proven itself within the existing Ford user base. The continued integration of TIES into Ford's design and development process will certainly lead to even more enhancements to TIES.

Acknowledgments

The TIES project is sponsored by a Ford committee, chaired by R. G. DeLosh and W. E. Jackson, that is involved in gathering technical information. Richard McWhorter and William Malinak of Inference Corporation produced the production version of TIES software. We thank the members of the TIES user community for their enthusiastic support. They have helped drive the evolution of TIES from concept to project to product over the last two years. We gratefully acknowledge the signif-

icant contributions of many individuals from both Ford Motor Company and Inference Corporation to the success of the TIES project.

References

Charniak, E.; Riesbeck, C. K.; McDermott, D. V.; and Meehan, J. R. 1987. *Artificial Intelligence Programming*. Hillsdale, N.J.: Lawrence Erlbaum.

Hauser, J. R., and Clausing, D. 1988. The House of Quality. *Harvard Business Review* May–June: 63–73.

King, B. 1987. Better Designs in Half the Time. In *Quality Deployment: A Series of Articles*, ed. A. Yoji. GOAL. Also in Japan Standards Association. 1986–1987. *Standardization and Quality Control*.

Steele, G. L. 1984. *Common Lisp: the Language*. Pensauken, N.J.: Digital.

Sullivan, L. P. 1988. Policy Management through Quality Function Deployment. *Quality Progress*, Special Issue on QFD: Listening to the Voice of the Customer, June.