

# Quantitative QFD for System Engineering

Mark W. Maier, Ph.D.  
EB-267

University of Alabama in Huntsville  
Huntsville, AL 35899

**Abstract.** Quantitative QFD (Q<sup>2</sup>FD) is an extension of conventional Quality Function Deployment (QFD). Q<sup>2</sup>FD incorporates quantitative requirements decomposition models into the QFD framework. It also rearranges conventional QFD matrices to facilitate quantitative capture and both forward and backward chaining of performance values. The rearranged matrices also allows full representation of QFD matrices with standard spreadsheets instead of only with special purpose software. Q<sup>2</sup>FD can serve as a bridge between quality, system, and disciplinary engineering personnel and act as an organizing formalism for all performance aspects of system design.

## SUMMARY

Requirements are typically divided into functional and non-functional. Functional requirements are behavioral requirements. These are the patterns of input to output processing the system must provide. Non-functional requirements—a very awkward term—include end-to-end performance specifications. These may include weight, cost, sensitivity, accuracy, and so forth. Many systems are driven by these quantitatively stated performance objectives. These systems may also contain complex behavior or other attributes, but the performance objectives are most important to the client.

For these systems it is common practice to take a performance centered approach to system specification, decomposition, and synthesis. A performance centered approach is especially appropriate when customer quality perceptions are driven by performance values, or where performance values are contractually specified. A particularly powerful way of organizing such a decomposition is through Quality Function Deployment (QFD) matrices. The QFD approach was pioneered by Japanese engineers looking for methods for carrying performance decompositions throughout the development process.

The Quantitative QFD (Q<sup>2</sup>FD) method extends conventional QFD to directly include quantitative performance models in the decomposition. It is a method for integrating quantitative, disciplinary engineering analysis in a single system engineering decomposition hierarchy. To integrate performance satisfaction models the method places the models into the cells of reformatted QFD matrices. The

reformatting allows full QFD matrices to be implemented with standard rectangular spreadsheets. It also makes use of the calculation power of computer spreadsheets.

A communication receiver example illustrates the various aspects of Q<sup>2</sup>FD. In the example we decompose specific performance objectives, like bit error rate, into top level engineering parameters. Q<sup>2</sup>FD captures their inter-relationship both graphically and quantitatively in a rectangularly gridded matrix. This matrix, which includes all the inter-relationship regions of the conventional, but non-rectangular, “House of Quality,” also allows back annotation of design values as requirements decomposition progresses.

## DISCUSSION

QFD is a Japanese originated method for visually organizing the decomposition of customer objectives [Hauser88, Akao90]. QFD could be viewed as a specific instance of other types of influence diagrams and matrices [Sage77]. Its intention is to provide a graphical hierarchy of how customer objectives are addressed throughout a system design, and carry the relevance of customer objectives throughout the process (potentially all the way to the factory floor).

A variety of authors have made use of the QFD framework to facilitate various aspects of total quality management [Bossert90, Day93]. Published approaches frequently make use of auxiliary matrices and combine with other quality tools. QFD, as usually described, is a semi-quantitative tool. The QFD matrices show numerical values for performance objectives, but those values are not automatically calculated from the QFD decomposition. The approach of this paper is to concentrate on the unmodified original matrix structure, but to add quantitative performance and quality metrics and reformat into a more convenient structure. Q<sup>2</sup>FD extends the QFD paradigm by dynamically integrating performance models. The method also makes the construction of QFD matrices more convenient by reformatting them to lie entirely on a rectangular grid. This facilitates use of computer spreadsheets instead of special purpose software. It also enables various advanced applications through directed search for optima, back annotation of evolving designs, and comparative tracking of distinct system configurations.

The discussion is organized to cover elements of quantitative requirements decomposition, a reformatting method for QFD matrices, and an integrated method. The first section lays out the Q<sup>2</sup>FD concept in brief. The next discusses quantitative requirements decomposition and its relationship to QFD. The next describes the visual formalism needed for Q<sup>2</sup>FD. This rectangular form is also applicable to conventional QFD if one wishes to use computer spreadsheets instead of special purpose QFD software. After an example of application of the method, we propose some possible advanced applications enabled by the Q<sup>2</sup>FD method.

**The Quantitative QFD Concept (Q<sup>2</sup>FD).** QFD, as usually described, documents a sequential decomposition of objectives into engineering or design parameters. The process begins with a set of objectives, which are measures of performance or quality of a system from the clients perspective. The QFD matrix documents a decomposition of those objectives into engineering design descriptors. The process is hierarchical with the engineering parameters at each level becoming objectives at the next level down. In principle, the process continues until the engineering parameters are directly implementable as parts specifications or even machine settings on the production line. The whole decomposition is documented in a chained set of QFD matrices, or “houses of quality,” with each linked to the next in sequence.

In Q<sup>2</sup>FD a “satisfaction model” which calculates a numerical value for each performance objective from engineering parameter values is embedded in the Q<sup>2</sup>FD matrix. By using several copies of the models it is possible to track several performance values. If system performance values are contractually required one set of values will represent the baseline decomposition to meet requirements. When the system has market competition a set of values should be used to track performance against known competition levels. A further set of parameters allows upward calculation of predicted performance from current engineering design values.

**QFD and Requirements Decomposition.** To understand QFD based requirements decomposition we first need a classification of performance requirements. Performance requirements differ dramatically in their analytical complexity. A simple decomposition into allocable, partitionable, and system or non-allocable serves the purpose here.

**Allocable Requirements.** An allocable objective is one which is met or addressed by a single component of the system. Such objective should be directly assigned to the performing component. Some examples include:

“The weight of the receiver WRA shall be less than 50 lbs.” This is assignable to the physical unit.

“*Objects on the display shall not rotate faster than 10 degrees per second.*” This is assignable to one functional process.

Because the correspondence between allocable objectives and design components is so clear, they tend not to produce serious problems even when there are very many of them. They do not require insightful system engineering because their consequences are evident. The next category, partitionable objectives, is more complex to handle but produces no greater level of conceptual complexity.

**Partitionable Requirements.** Partitionable objectives are those which can be converted into allocable objectives in a single, simple step. Typically they are requirements whose partitioning requires only dividing the required value into values of the same dimension among objects in the functional and physical models. They may be captured in the QFD matrices with the non-allocable objectives if complete traceability in a single place is desired. Some examples are:

“*The system weight shall not exceed 400 lbs.*” This objective can be divided by assigning weights to each architecture module such that the sum is less than the required value.

“*From the receipt of message XX to the display of YY the elapsed time shall not exceed 500 msec.*” In the first step this objective can be decomposed into sub-timing objectives for components. The decomposed quantities are all of the same dimension (time) and are combined in the simplest manner (additively). An objective may be partitionable at one level of system modeling, but a piece may be a more complex modeling problem at another level. For example, timing requirements where implementation dependent considerations like tasking and resource allocation appear.

**System Requirements (Non-Allocable).** Non-allocable requirements are the heart of a true system specification. These are requirements which are not met by any one portion of the system and which must be decomposed in multiple steps into sub-requirements of varying dimensionality. They describe emergent properties of a system. Some examples are:

“*The detection range shall exceed 50 nm for each of the defined target types.*” Detection range is met only by the interaction of all portions of the system and is specified by the combination of dissimilar parameters like system noise figure, detection bandwidth, threshold SNR, antenna gain, etc. Moreover, many of these parameters must themselves be decomposed before they can be assigned to an object in the functional or physical models.

*“The probability of correct identification of intercepted signals shall be at least 90%.”* This objective likewise must be decomposed into dissimilar sub-objectives which themselves may require decomposition before allocation to the functional or physical models.

How does the system engineer address the true system level requirements? Analytical design for such requirements requires satisfaction models. A satisfaction model is a model, drawn from the design disciplines relevant to the system, which predicts performance for the system requirement in terms of lower level engineering design parameters. A satisfaction model for the detection range example above will include the parameters mentioned, threshold SNR, antenna gain, noise figure, etc. Some of these parameters may be allocated directly, which others are themselves "system level" requirements for subsystems.

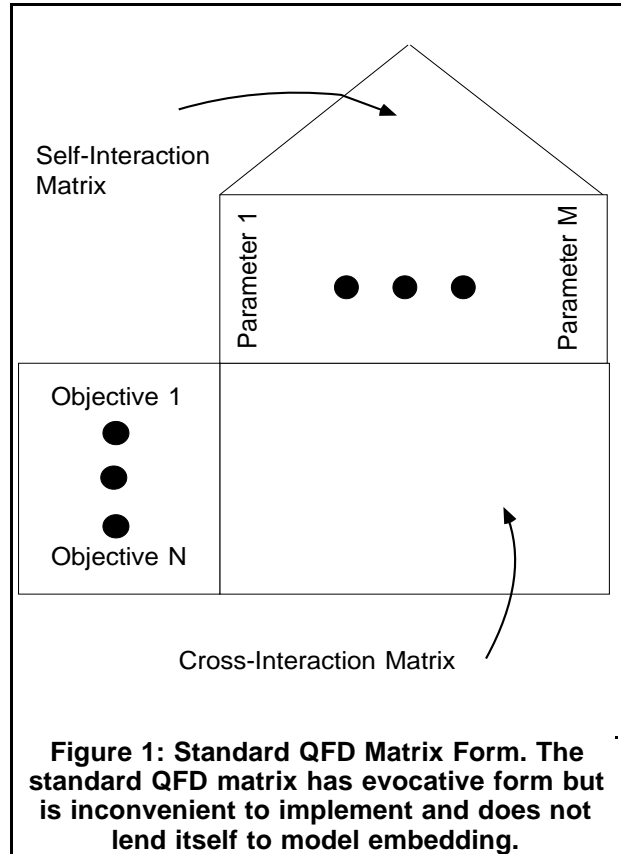
The analysis of objective categories leads to additional issues in how functional or behavioral requirements and quantitative performance requirements interact. This subject has been addressed in more detail in [Maier96].

**The “Basement Matrix” Form.** Traditional QFD uses the “House of Quality” matrix as a visual formalism. The house is illustrated schematically in figure 1. This matrix form has several important drawbacks. First, its non-standard grid structure and text alignments requires specialized software packages to represent. These packages are often expensive, of limited availability in engineering organizations, and have limited functionality. Second, the non-standard nature of the packages makes them much harder to integrate with existing analysis tools. An alternative form, shown in figure 2 is well adapted to computer spreadsheets and engineering analysis.

The standard QFD matrix contains four principle parts. The left box, labeled Objective 1 to N, contains a listing of system objectives or requirements. The terms are used interchangeably here, with the distinction one of the degree of importance placed on specific performance levels by the customer. These are given, at the top level, in the customers language insofar as possible. The box across the top, labeled parameter 1 through parameter M, contain a listing of system design parameters that relate to the objectives. It should be possible to compute satisfaction levels for the objectives given a set of values for the parameters. Put another way, the objectives are performance metrics for the system and the engineering parameters are those system characteristics which determine performance levels.

The box in the center, labeled Cross-Interaction matrix, is a visual indicator of the connection between the objectives are parameters. Visually evocative markers are used in the cells in the region to indicate the strength and direction of the linkage

between an objective-parameter pair. Typical values could be strongly negative, negative, neutral, positive, and strongly positive. The triangular region at the top uses the same markers for strength of connection to show the connections among the parameters. The parameters are not generally orthogonal, a change in one system design parameter usually implies effects on other parameters.



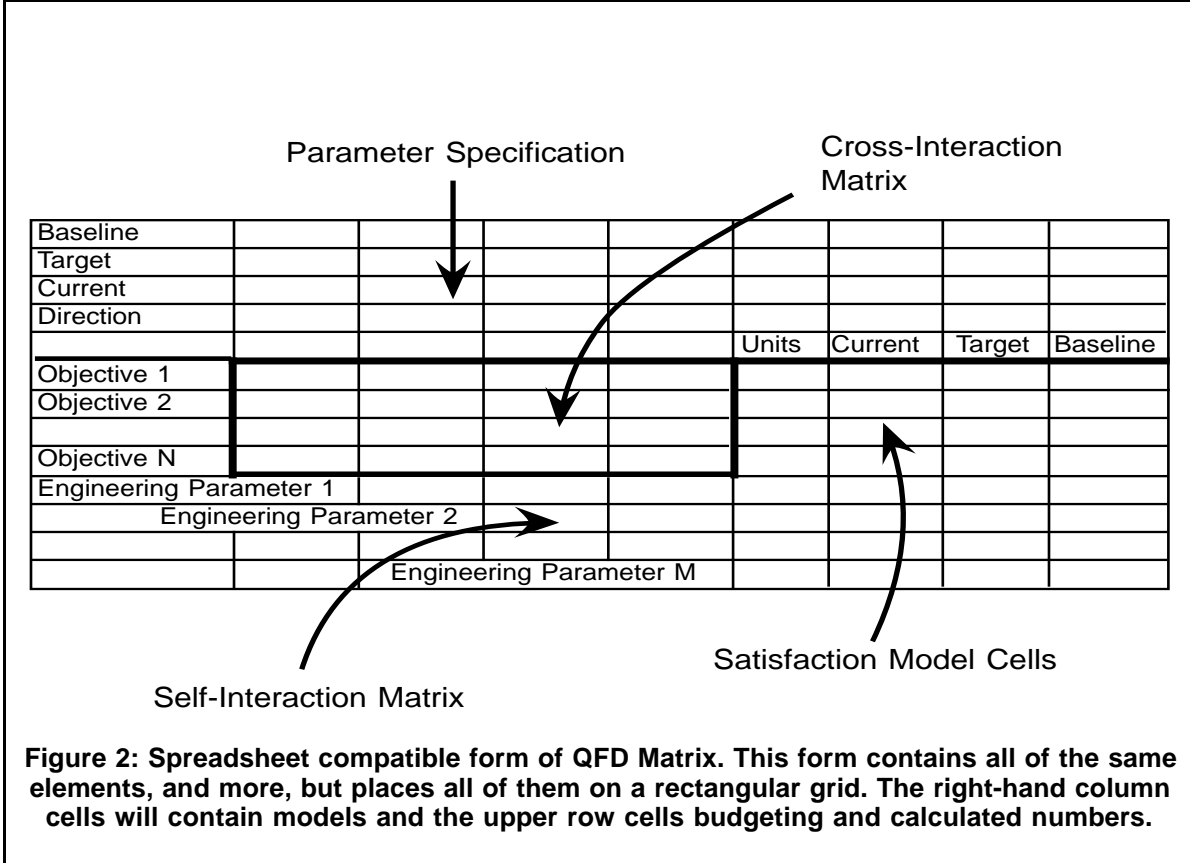
Further information can be added to the right of the cross-interaction matrix. The region on the right can hold performance values for the objectives calculated from various parameters levels. The parameter values can be placed along the bottom of the cross-interaction matrix. As an example, right hand columns might be established for a competitors performance values, targeted performance values for the system under design, values reflecting the nominal parameter budgets, and values reflecting the current state of the system design and lower level estimates for achievable parameter performance. In a similar fashion, the area below the cross-interaction matrix can hold multiple values for the parameters. Possible choices include design budget values, targets, and values calculated upward from lower-level, more detailed design models.

By carrying calculated and budgeted values, and by chaining among matrices, they become a vehicle for visibly extending customer objectives to low design levels. In a QFD ideal, each objective is traceable in QFD matrices through a chain of parameters down to the lowest operating level. In principle, the low level is individual part

specifications and manufacturing tolerances and machine settings. This forms a grand chain of traceability that allows the impact of all design choices to be seen on customer objectives.

An alternative form for QFD matrices is shown in figure 2. This form resets the QFD matrix elements on a horizontal grid to facilitate use of widely available and low cost computer spreadsheets for QFD.

refinements of parameters can be chained to the right and down in additional matrices. The original objectives that form a vertical column on the left can also be arranged in a triangle for complete symmetry. This complete, triangular form for multiple, chained QFD matrices on a rectangular grid is illustrated in figure 3. Modern spreadsheets contain extensive analytic modeling power. Cells can invoke programs of very powerful internal functions, or custom



**Figure 2: Spreadsheet compatible form of QFD Matrix. This form contains all of the same elements, and more, but places all of them on a rectangular grid. The right-hand column cells will contain models and the upper row cells budgeting and calculated numbers.**

This alternative form provides spaces for the objective-parameter interaction matrix in the central rectangular region and the parameter interactions in the triangular region on the bottom. The reader can quickly verify that the right triangle on the bottom is exactly equivalent to the peaked roof matrix in the traditional QFD matrix. Since this form is on a rectangular grid it can be embedded into a standard computer spreadsheet. Figure 2 was drawn in Excel and the author has built large QFD models in this manner.

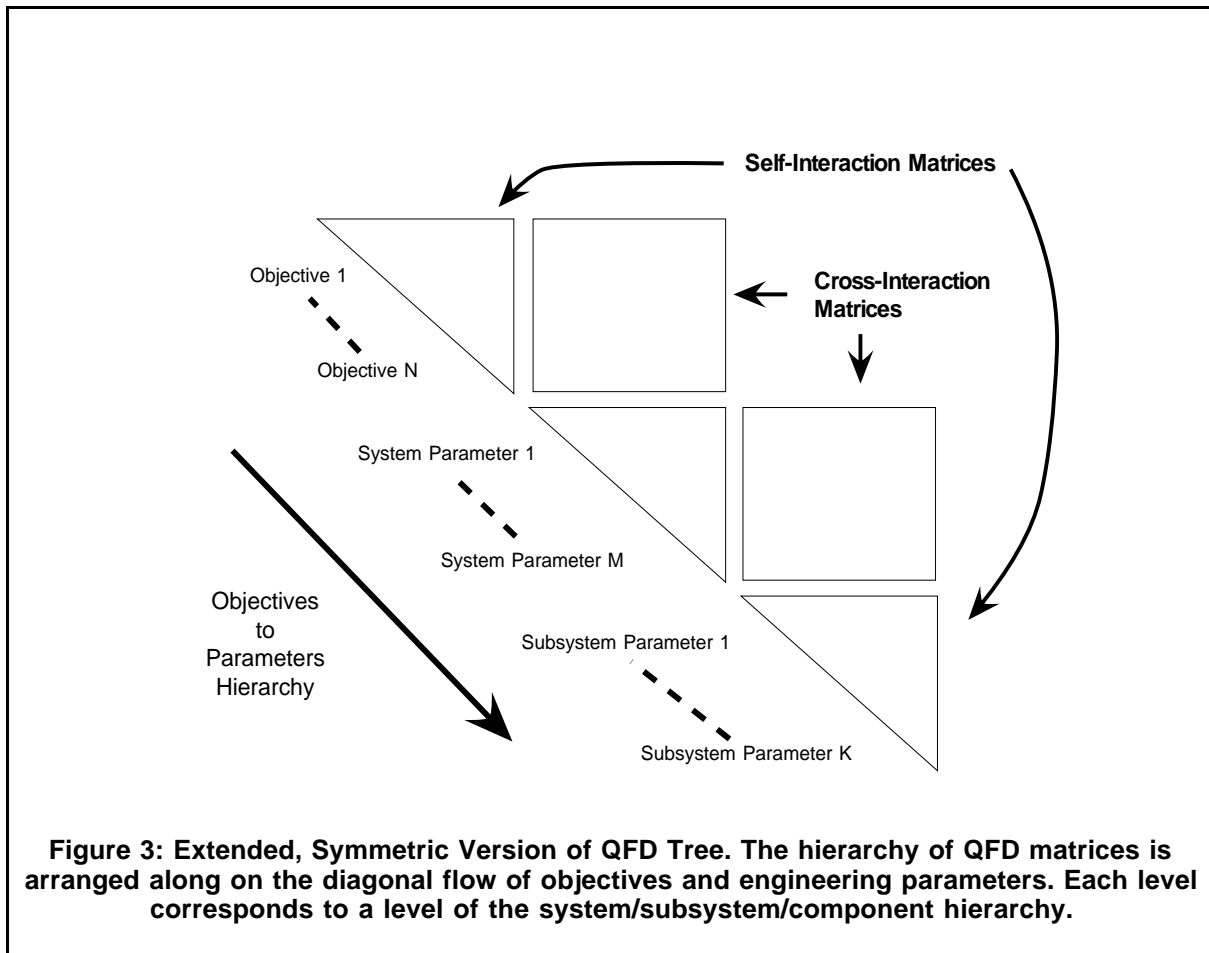
The horizontal format makes clearer the connection among hierarchies of QFD matrices. QFD matrices can be linked in hierarchies with the parameters at one level becoming the objectives at the next. This chaining is very much in the fashion of the system engineering adage that one engineers design becomes another's requirements.

Computer spreadsheets allow easy chaining of matrices and consolidation of numbers over multiple matrices. The engineering parameters along the bottom of the matrix can act as the objectives for the next level and directly feed another cross-interaction matrix off the figure to the right. Successive

external software. Some spreadsheets are designed to interface to other analytic tools and to act as the centralized consolidator of modeling.

This alternative form also lends itself to the tracking of performance allocation. The cells in the right hand columns of figure 2 can be hard coded, or computed from the values in the rows along the top of the matrix. For example, the row and columns labeled "current" can be computed and continuously reflect the status of the design in a process referred to as back annotation. The right hand "current" column values would be computed from the "current" row. The values in the current row would themselves be computed from current values of engineering parameters from lower levels in the QFD hierarchy. The tree of computation would terminate at the level to which the design has progressed.

Using this methodology, as current values change, their impact is immediately reflected all the way to the original customer objectives. Additional columns and rows can be devoted to tracking other sets of parameters values, such as targeted, specified, competitors, etc.



**An Example.** Consider a simple digital data receiver required to operate with interference on adjacent channels. Typical receiver objectives include:

1. Maintain a bit error rate below 1 per minute.
2. Receive the data signal at as large a range as possible.
3. Maintain bit error rates in the presence of radio signals in neighboring channels.

Beginning with semi-quantitative statements, form quantitative objectives. The first becomes the bit error rate, the second the signal reception range, and the third the amount of co-channel power tolerable. All three are referenced to each other. The range must be defined so that the bit error rate is specified at the signal level corresponding to that range. Likewise, the co-channel power rejection should be the maximum co-channel power level that has minimal impact on bit error rate. We have to add external parameters that define the environment, in this case the transmitter power we are trying to receive and its bit rate, and suitable engineering parameters for the top level design. Here we select (each followed by its abbreviation in parentheses):

- Antenna Gain (G)
- Noise threshold (NT)

- Receiver Noise Figure (F)
- Receiver channel filter bandwidth ( $BW_{RF}$ )
- Channel filter stopband rejection ( $F_{rej}$ )

Abbreviate the transmitter equivalent radiated power as  $ERP_T$ , standard noise power density as  $kT$ , and bit rate as  $R_b$ . A simple, but reasonable, set of satisfaction models is:

$$BER = R_b \operatorname{erfc}(\sqrt{NT})$$

,  $\operatorname{erfc}(x)$  is the complementary error function

$$Range = \sqrt{\frac{GERP_T}{4\pi NT(FkT BW_{RF})}}$$

$$\text{Co - Channel Power} = F_{rej} + NT$$

These models are embedded in an example QFD matrix shown in figure 4 in the right hand columns. The environment parameters are listed at the bottom for convenient access. Many procedures can be used to fill in the target and current values. One procedure is:

1. Set objective values in the column labeled "target." These are set by study of customer need/demand, competitive analysis, estimated feasibility, etc. These values are hard coded.

2. Embed the satisfaction models in the cells of the columns labeled "target-calc" and "current." Then set values in the row labeled "target" and observe the calculated values of the objectives in the "target-calc" column.
3. Iterate the target row values until the values calculated in the target-calc column are acceptable.
4. Repeat the process for the values in the row labeled "current."
5. Extend the analysis to the next level down in the hierarchy. This next level down will calculate values for the upper level's engineering parameters. These calculated values should replace the hard coded values in the "current" row. That row, and the associated column, now reflect the design as it is currently.
6. Additional rows and columns can be added to track budgets versus targets, minimum contractual requirements, etc.

As target values are varied, or current values evolve, their impact on customer objectives is directly displayed. Problems become evident as they happen, and the large scale consequences of low-level design decisions are directly displayed. Moreover, the analytical capabilities of the spreadsheet can be used to conduct design explorations and optimizations.

The application of this methodology is clear where reasonably simple satisfaction models exist. As the bit error rate example shows, even fairly complicated functions like the complementary error function are available in spreadsheets. But what about the case where no simple model exists? Perhaps the only model for the satisfaction of an objective is a simulation or even expert judgment.

For example, if production cost were a

requirement it might be available only from human estimates based on fragmentary designs. Bit error rate in a complex communication system can be calculated only through simulation. In complex, socio-technical systems (for example Intelligent Transport Systems) many of the objectives are quite subjective.

In these cases we can use the matrix with a mixture of dynamic calculation and hand coded entries. For example, append a new row for a cost objective to the matrix of figure 4, append any new parameters desired to cover additional cost relevant parameters, and hand enter the best estimate of cost for the target and current parameter values. The other objectives with explicit models will continue to be calculated. Of course, when the entries are manual it is much easier for them to get out of synchronization with lower level parameters. Software tools could track change dates and determine when manual values had gotten out of date. More detailed examples of the use of QFD with quantitative requirements models have been given in [Maier95].

**Advanced Applications.** The Q<sup>2</sup>FD method enables a number of advanced applications. Three extensions of particular interest are back annotation, automated search for optima, and linkage derivation. All rely on the revised structure of the matrices and coupling with computer databases and tools.

**Back Annotation.** Back annotation refers to tracking current engineering design values upward through the decomposition hierarchy. The object of back annotation is to make the impact of detailed design decisions continuously visible in terms of customer requirements. While purely top-down decomposition is an ideal, in reality detailed designers and vendors will often either fail to meet top-down decomposed parameters or will be able to substantially exceed planned parameters. Either case should be tracked since detailed design failure may

Variation	2	3	3	1	10					
Target	0	15	7	10	60					
Current	-3	12	10	10	50					
Direction	+	+	-	-	+					
	dB	dB	dB	MHz	dB	Units	Dir	Current	Target	Target-Calc
Bit Error Rate		↑				/sec.	-	9.01E-03	1.00E-05	8.88E-10
Range	↑↑	↓	↑	↑		km.	+	793	1000	1120
Channel Power		↑			↑	dB	+	62	60	75
Antenna Gain										
Noise Threshold										
Receiver Noise Figure				↑						
Receiver Channel Filter Bandwidth										
Channel Filter StopBand Rejection										
ERP_T	20 dBw									
kT	4E-21									
R_b	1E+06									

**Figure 4: QFD Matrix for simple receiver example. The alternative form captures the interaction indicators and the quantitative models in a single rectangular format matrix.**

cause system level failure and detailed design unplanned success may be able to compensate for other problems or provide new opportunities.

The back annotation concept has been described in the preceding sections. Back annotation is accomplished by flowing quantitative information on engineering parameters upward from lower level QFD matrices to the upper level matrices. At each level of upward flow the parameters are used to calculate predicted values for the customer objectives. Thus low, detailed level design information is continuously reflected in performance or quality estimates as detailed design proceeds. For example, as known part performance levels become available their impact on predicted performance percolates upward.

If multiple budgeted and assigned objective values are tracked, as shown in the previous figures, a continuous comparison between desired and predicted performance can be made. The QFD matrices become a framework for downward allocation (a form a traceability) and upward monitoring (the necessary companion for traceability).

If the QFD matrices are to be used for extensive back annotation, some means of connecting the detailed design parameters values to the low level matrices is needed. Of course, it could be done manually with design values hand entered into QFD spreadsheets. Current spreadsheets can be configured to perform consolidation across many machines distributed around a network, which would facilitate back annotation. A better solution would be to have the QFD matrices draw their data from design databases. In principle this is possible anytime the design data is stored in open formats in design tools. In practice it may be quite difficult since design database formats are quite diverse and often proprietary.

**Optimal Search.** When quantitative satisfaction models are in place the capabilities of modern spreadsheets can be exploited for search. Possibilities include search for feasible regions within a constraint space and optimization of selected requirement values. Search is conducted in the framework of the satisfaction models. Take the case of figure 4 as an example. Given the satisfaction models for the three objectives in terms of the parameters, what set of parameters values should be chosen? In optimization terms one would need a utility function for customer preference expressed on the objective values. If the objectives were complete (which figure 4's are obviously not since they do not include cost) one could then ask what set of parameters produce the highest utility. Alternatively, if the objectives are given as requirements, the question would be what is the set of parameters values which meet or exceed the required objective values.

Such optimization is not new, it is design optimization in a general guise. What makes it interesting in the QFD framework is twofold. First, by introducing the quantitative models and embedding

them into computer tools the use of automated computational search tools is facilitated. Many computational goal seeking tools are now being developed for spreadsheets because of their wide use in developing financial and engineering models. Second, the interaction matrices can provide guidance in searching. The directions and magnitudes suggest which parameter variation directions are important. If one wants to manipulate a particular objective, the matrices indicate the network of relationships among other objectives and the associated parameters. At least in principle, a variety of interaction and influence diagrams could be extracted from the interaction sections of the matrices.

**Linkage Derivation.** In the reverse fashion the satisfaction models can be used to derive the linkages in the inter-relationship cells. The satisfaction models contain all the information needed to determine the interaction relationships. The parameters used in a particular objectives satisfaction model are the parameters which influence it. The derivative of the satisfaction model with respect to the parameter indicates the strength and direction of the interaction.

The models can be used to derive the interaction symbols. Starting from a nominal design point, small variation calculations will yield numerical derivatives for the requirement values in terms of the engineering parameters. Thresholding and assignment of difficulty weights then yields the linkages. Of course, considerable judgment is required to carry out this process. The strength of the interaction must be referenced to some measure of how difficult (in an engineering sense) it is to move a particular parameter. The satisfaction models do not contain that information. Only knowledge of the technological context can determine the strength. The direction must be considered carefully, since positive and negative correlations are likewise subjective. Positive and negative movement directions are presumably clear for the objectives, positive movement is in the direction that increases customer satisfaction with that objective. For the parameters the notion of positive and negative must be defined. It could be taken just a smaller and larger, or mapped to some notion of "better" and "worse" for the particular parameter.

## CONCLUSIONS

QFD is a powerful tool and can be made considerably more powerful through quantitative extension. With the quantitative extensions it becomes a central organizing formalism for system engineers for the performance decomposition of their work. QFD is not a tool for documenting functional requirements decomposition, but the two can be linked.

To use QFD as such an organizing tool, two extensions were required. First, embedding quantitative satisfaction models that relate customer

satisfaction of particular objectives with engineering design parameters. Second, the reformatting of the QFD matrices (in this case rectangularly) to facilitate the use of computer tools. By placing the matrices in a flexible tool, preferably linked to a database, the QFD system becomes suitable for forward budgeting and allocation of customer objectives and back annotation of detailed design as it progresses.

#### REFERENCES

- Akao, Y. (editor), *Quality Function Deployment: Integrating Customer Requirements into Product Design*, ASQC Quality Press, 1990.
- Bossert, J. L., *Quality Function Deployment: A Practitioner's Approach*, ASQC Quality Press, 1990.
- Day, R. G., *Quality Function Deployment: Linking a Company with Its Customers*, ASQC Quality Press, 1993.
- Hauser, J. R., and Clausing, D., "The House of Quality," *Harvard Business Review* 66, No. 3, pp. 63-73, May-June, 1988.
- Maier, M. W., Quantitative Engineering Analysis with QFD, *Quality Engineering*, 7:4, pp. 733-746, 1995.
- Maier, M. W., Integrated Modeling: A Unified Approach to System Engineering, *Journal of Systems and Software*, 32:2, February. 96.
- Sage, A., *Methodology for Large Scale Systems* (New York, NY: McGraw-Hill, 1977).

#### AUTHOR BIOGRAPHY

**Mark W. Maier** received the B.S. and M.S. degrees from the California Institute of Technology and the Engineer and Ph.D. degrees in Electrical Engineering from the University of Southern California. While at USC he held a Hughes Aircraft Company Doctoral Fellowship, where he was also employed as a section head. Currently he is Assistant Professor of Electrical and Computer Engineering at the University of Alabama in Huntsville. Dr. Maier's research interests are in randomized radar waveforms, data compression, computer based system design, and system architecting for large scale systems.