USING QFD TO TAME WICKED INFORMATION SYSTEM'S REQUIREMENTS

Steve Howard
Swinburne CHI Laboratory, School of Information Technology,
Swinburne University, PO Box 218, Hawthorn, 3122. Australia.
showard@swin.edu.au

ABSTRACT

This paper argues that information system's requirements can be in tension with both themselves and the context to which they apply. The tensions act as limiters on our ability to deliver effective information systems. In the paper we describe examples of such limiters. The paper then presents a response to this problem in the form of Quality Function Deployment and calls for a more systems oriented view of information systems development, which though not capable of removing the ceiling, is capable, in principle, of engineering best compromises. A case study is used to illustrate the application of QFD to the specification of information system's requirements.

Keywords

Requirements, QFD, House of Quality, Trade-off analysis

INFORMATION SYSTEMS, COMPATIBILITY AND WICKED PROBLEMS

Ensuring compatibility between the various elements of an information system, eg. users and their tasks, organisational structures and culture, and information technology, has reigned as the primary goal of Information Systems for the past two decades (Avison and Fitzgerald, 1988). Even assuming that we can design for compatibility, what do we do when two (or more) different 'compatibilities' are in tension and we, as information system designers, cannot ensure both? Information system compatibility, when seen from this perspective, is a 'wicked' problem (Rittle and Webber, 1973 and Vicente et al, 1997) characterised by:

- The lack of a definitive formulation. Different stakeholders have different perspectives on what is and what is not the most important information systems issue;
- Solutions to information systems design problems tend to be more or less satisfactory, not 'right' or 'wrong';
- The solution space to any information systems design problem is not exhaustively identifiable. Often candidate solutions are overlooked, less satisfactory solutions chosen etc.

The traditional response to the problem of information systems requirements analysis is essentially reductionist and based on principles of decomposition or step-wise refinement. Though suitable for complex and well defined problems (where the means-ends relationships are stable and understood) such an approach is problematic when we consider the characterisation of wicked problems presented above. This paper builds of many traditions: Churchman's (1971, '78) work on Inquiring Systems; Stowell and West's (1989, '90) work on client-led design; Checkland's impressive work on the soft systems methodology (Checkland, 1991); and work by the Quality Function Deployment community (Cohen, 1995, Akao, 1990) on capturing and using the 'voice of the customer' in product development. We present a discussion of the problematic of wicked requirements problems and a practical tutorial to a technique that can be used to analyse such complex requirements situations. The paper will be of most interest to those unfamiliar with the techniques of Quality Function Deployment and those tasked with the analysis of wicked requirement situations.

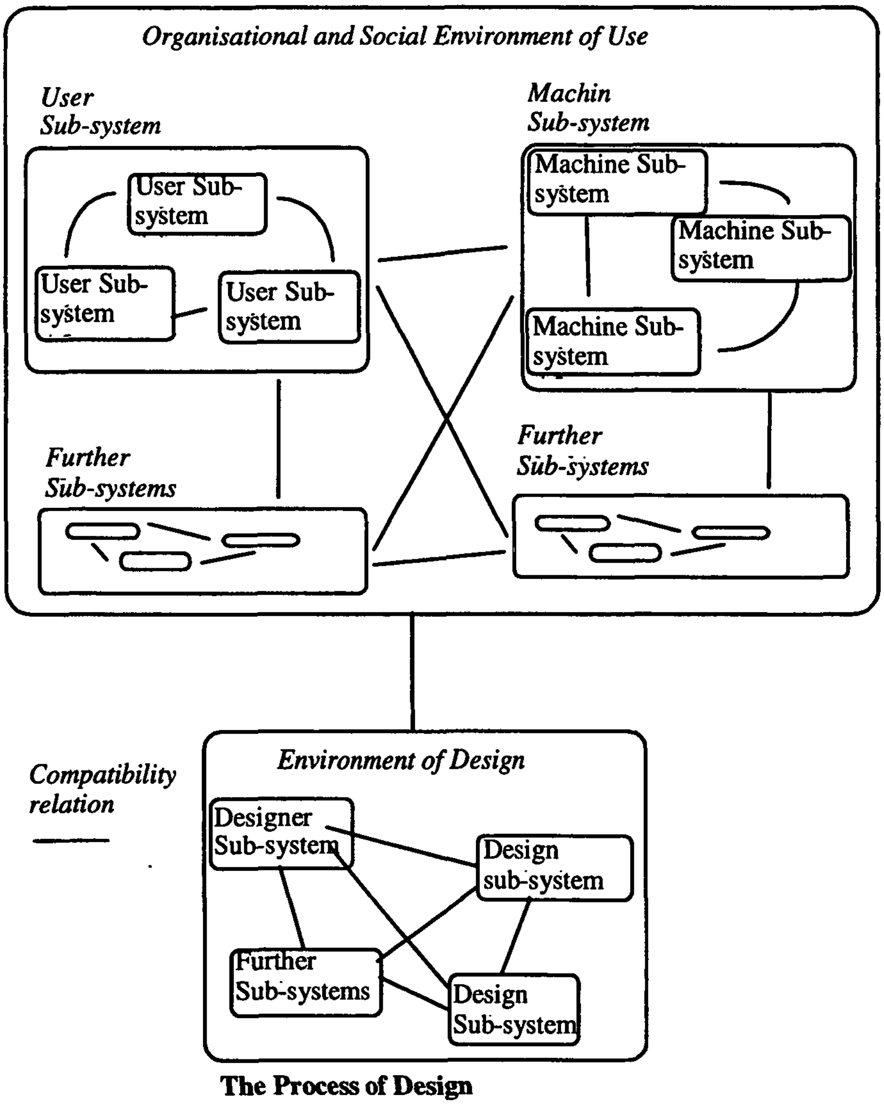
Consider an example from another domain for illustration. 'The Australian Iron Man' competition is a triathlon composed of run, swim and board (essentially a sea kayak) legs. In the early 90s, Trevor Hendy (a previous Australian Iron Man champion) attempted to become a member of the Australian Olympic Kayak Team. To do this, Hendy preferred board training over run and swim activities. During the early part of 1996 Hendy returned to Iron Man competition and struggled in the swim and run legs of the event due to the combined effect of underdeveloped lower body strength and increased upper body muscle bulk. There would seem to be a tension then between the three individual sports and therefore a conflict when sports are combined within one individual. Iron men must compromise their performance in the individual sports in order to optimise their performance overall. It follows then that it is not necessarily the case that a given system (in this case a triathlete) can be designed with maximum compatibility between all of its subsystem components (i.e. maximum performance across all three sports). It may be appropriate to sacrifice sub-system compatibility (e.g. board, swim or run leg performance) for the sake of overall system goals (e.g. triathlon performance).

Do such tensions exist in information systems design? Figure 1 illustrates an information system as a set of 'compatibility relations'. It should be stressed that the diagram is not intended to be complete, on the product side for example one could reasonably add the user's task or, in the case of a CSCW system, other users. It should also be stated that the compatibility relations illustrated are those seen from a particular perspective, and are not independent of that perspective. The technique we go on to present below can be used equally well in a relativist mode where the final expression of the requirements depends on the discourse between various stakeholder groups (and so 'optimum conpatibility' would be that defined by a particular group) as in a positivist mode where the requirements are a function of the implied system's ontology.

In Figure 1 the design problem is seen as one of 'information system design'- that is, designing the most appropriate fit between all of the elements in the information system (including user, organisation, technology etc) rather than simply software or hardware design. The user can be seen as consisting of a number of subsystems (e.g. visual, motor, cognitive...), and likewise the machine can be viewed in terms of its elements (input dialogue, output dialogue, input and output devices etc). In turn, the subsystems themselves can be broken down into ever-smaller units, for example a mouse consists of various mechanical and electrical components. An information systems designer might be concerned with the compatibility between screen layout and the task; between the amount of domain knowledge possessed by the user and that required by the computer; between the computer system and the organisation within which the users work etc. Unpacking the diagram further, the designer also has to consider the environment within which the information system will be situated and the impact that the physical and social context will have over the information system. These seem like obvious points, but the implications, developed below, are profound.

Figure 1 Compatibility Relations and Information Systems

The Product of Design- Information System element compatibility



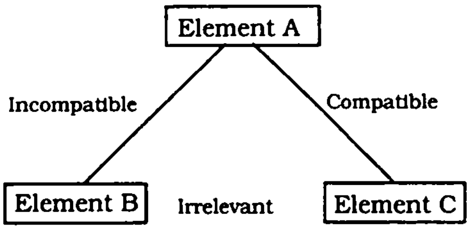
The Process of Design

Potentially then, all information system subsystems can be in a compatibility relation with all other subsystems- the number of possible compatibility relations being very large indeed for most real systems. There are three possible logical relations between information system subsystems: irrelevant, compatible and incompatible, as shown in Figure 2.

Irrelevant relations exist when two information system subsystems are not connected (ie. do not communicate and are not interdependent), as in the case of elements B and C in Figure 2 above. For example, a designer may believe that a mouse is an appropriate input device for the design task in hand, and that an information system should model domain knowledge. These two subsystems (input device and domain model) are not connected and are therefore irrelevant.

Compatible relations exist when acting on all alternatives is possible, as is the case where a designer decides to implement both a menu and a command based interface and allow information system users to select between the two (interestingly, the users then would be forced to make the selection).

Figure 2 Three Compatibility Relations



Incompatible relations exist when a clear choice must be made between the alternatives, when for example a menu or command based interface must be selected. In this case, the alternatives are clearly in tension. The next section will consider examples of incompatible relations drawn from ongoing Information System research.

Irrelevant relations are significant because they can become compatible or incompatible relations. If the designer above decided to implement both menu and command interfaces at a risk to the project deadline the two irrelevant alternatives are made incompatible.

Any given information system can therefore be seen in terms of a network of interrelated compatibility relations- relations between and within the user, the technology and the organisational and social environment- at a number of levels of abstraction. The implications of such decomposition however challenge the very notion of being able to 'design for compatibility' for as we have seen, there are in fact many 'sub-compatibilities' and when we have many of anything there is the possibility that they might interrelate and compete. Put simply- the various compatibility relations within any information system, that must be adequately engineered if we are to have successful information systems, can be in tension with each other. Further, Figure 1 alludes to the fact that the information system product can also be in tension with the design process- when for example the designer does not have appropriate resources (tools, knowledge, time) to deliver a desired information system.

Any real design problem will contain many such compatibility relations and, given their interrelated nature, design might be better conceived as being concerned with achieving the most appropriate balance, rather than striving for an abstract ideal information system. In part this is another way of saying that the design of information systems is a wicked problem.

One way to reason about the information systems, and their composition from interrelated compatibility relations, is to distinguish between harmony and compatibility. Harmony is a global compatibility, and describes the degree of dissonance between *all* subsystem elements, including users, tasks, organisational contexts etc. In this formulation, systems can vary between maximum harmony, which represents the presence of complete compatibility between subsystems, and minimum harmony, which represents the situation of complete incompatibility between these elements. All information systems can, in principle, be placed somewhere on this continuum. The goal of information systems development is to create a system with maximal harmony, though in practice, due to the tensions discussed above and below, satisfactory harmony only is possible. In order to identify the degree of harmony embodied in a system, it is necessary to consider the various compatibility relations, ie. sub-system elements and their interrelationships. The measure of harmony then becomes the total accumulation, or aggregate, of system compatibility relations. As illustrated above however, compatibility relations can take three forms (irrelevant, compatible and incompatible) and this will affect the derivation of harmony. The next section illustrates situations in which compatibility relations are not simply additive, but competitive, in their contribution to compatibility.

INFORMATION SYSTEM'S REQUIREMENTS CEILINGS

The examples below illustrate different types of tension in various information systems. Of course, by judicious design, it may be possible to resolve the tension and 'find a way out'. We do not mean to imply that the tensions are insurmountable, only that they are present and, given appropriate tools (see 'House of Quality' below), perhaps resolvable. The tensions fall into three classes- tensions between information system subsystems, tensions between information system subsystems and their context and tensions between development processes and the information system. Each tension can be seen as the tension between two or more compatibility relations in their contribution to harmony.

Between different subsystems

- ***Individuals versus groups and time-shared computers***
  Time-sharing on main-frame computers satisfies the requirements of a group sharing the same resource with a minimum of encumbrance, yet does not satisfy the requirements of an individual programmer in terms of compile time, response time etc. Designers of such systems have made a choice to sacrifice the needs of the individual programmer, so as to optimise the availability of resources for the group. How do we represent such choices and allow information systems design teams to reason about the consequences of those choices?

- ***Real time versus batch and order entry***
  Users taking transient orders (e.g. over the phone) often require different functionality of computing support than users inputting orders from paperwork due, in part, to differences in the temporal aspects of their tasks. How should designers respond to this challenge when both groups of users are to use a single system?

- ***Experts versus novices and the user interface***
  Users with differing computer expertise may have different requirements of computing support e.g. command input versus menu selection, access to advanced features, access to on-line tutorials. Information systems designers can respond to this challenge in a number of ways, e.g. ignore the variation and produce a single interface, produce user-centred configurable interfaces, produce machine-centred adaptive interfaces. How do we support the designers and users in exploring and prioritising the options?

- ***Consistency versus compatibility and sub-optimal style guides***
  Many information system designers are required to follow standards and guidelines. However, many standards do not embody optimal advice (Teasley and Scholtz, 1997). Here we see a tension between adhering to the style guide to ensure consistency across information systems and ignoring the style guide so as to ensure compatibility between the user and the information system. Is it better to be consistently incompatible or inconsistently compatible?

- ***Abstract versus concrete and corporate style guides***
  Ongoing work in the implementation of Corporate Style Guides aims to provide designers with concrete instantiations, or exemplars, of the advice contained in the style guide. However, any particular abstract style rule can be instantiated as any number of concrete exemplars. Furthermore, concrete exemplars may arbitrarily and unwisely constrain the solution space. Providing exemplars is 'good' (they make concrete the rules of design), but providing exemplars is also 'bad' (they may restrict the cognitive solution space to that bounded by the example). Do we use exemplars or not?

In the above examples, the conflict can be seen as a 'subsystem ceiling', where information system subsystems themselves present a ceiling to their optimisation. Improving one compatibility relation necessarily negates another, unless the design problem can be reconceptualised. If a subsystem tension cannot be designed out of the system then negotiations to achieve compromise will be necessary between the relevant subsystem owners. A subsystem tension then is generally resolvable at the level of the subsystems concerned.

Between subsystems and their context

- ***Message versus context and headphone use***
  Current research (e.g. Petrie et al, 1997) in navigation aids for visually impaired people has struggled with the tension between providing users with a clear auditory signal (by requiring, for example, users to wear headphones) whilst not isolating users from the surrounding, and potentially dangerous or otherwise significant, physical and social context. The challenge here is that both a clear auditory message and access to environmental cues are important components of the information system and should be optimised, however they are in tension. To conceive of the problem as one of simple information system compatibility is useless when the very compatibilities themselves are in tension.

Here the tensions can be seen as context ceilings, where the context presents a ceiling to the optimisation of subsystem compatibility. If context ceilings cannot be side-stepped through elegant redesign then they imply high level changes may be required involving many stakeholders. Identifying the type of tension is important in understanding the nature of the appropriate resolution.

Between product and process

- ***Usability versus cost***
  Usability is often cited, sometimes unfairly, as a cost factor in the development process (Bias and Mayhew, 1994). 'We could produce more user-friendly systems, but we just don't have the money or the time' is the cry often heard. Whether real or apparent, usability (and other information system product qualities) and development costs frequently seem in tension. A designer capable of resolving this tension needs to know about development budgets and usability factors, and needs to understand the knock-on effects that each has on the other.

Here the conflict can be seen as a process ceiling, where the development process itself presents a ceiling to the optimisation of usability. Process ceilings are common in information systems development; the system cannot be optimised because the budget has been spent, or the time has expired, or the team do not have the relevant expertise, or access to the right sort of stakeholders could not be obtained. If process ceilings cannot be avoided through appropriate design then negotiations between the customer and the developers will be necessary, aimed at changing either the product requirements, or the project constraints, or both.

Summary

The three compatibility tension types illustrated above are not necessarily mutually exclusive. Complex design problems will often contain many tensions. Further, there are semantic difficulties in identifying types of tension, in particular, whether a problem is seen as a subsystem tension or a context ceiling depends in part on the grain of analysis used, in that other subsystems themselves occupy the environment of any particular subsystem.

In each case above, if the tension cannot be removed by reconceptualising the design problem then an appropriate balance must be struck involving negotiations between the stakeholders and developers cited. What then can we offer the identification and satisfactory resolution of such tensions?

THE HOUSE OF QUALITY

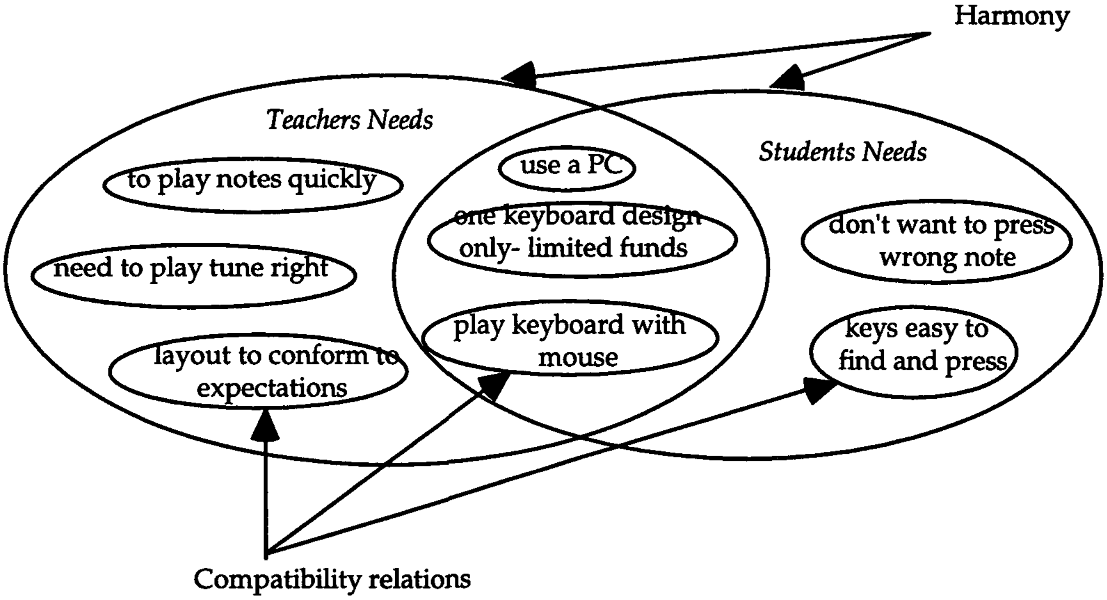
Quality Function Deployment (QFD) is a systematic planning process designed to explicitly incorporate the 'voice of the customer' into product design. Used in the manufacturing industries (Akao, 1990, and Cohen, 1995) to prioritise customer needs and map these needs onto technical constraints, applications are appearing in the software domain (Shindo, 1991 and Zultner, 1992).

QFD's primary components include a structured planning process which draws input from cross-functional teams working within a concurrent engineering paradigm. Ongoing decision making and communication is supported through the use of a graphical notation termed the 'House of Quality'. Although aspects of the planning process are hardware specific (Cohen, 1995), the House of Quality notation appears to work well for software development (Shindo, 1991 and Zultner, 1992, Ala-Siuru, 1993 and Lundell and Williams, 1993). This section considers the use of the House of Quality to articulate and reason about the types of tensions discussed above.

Planning and Gathering the Voice of the User

The first step in developing the House of Quality (HoQ) involves collecting the 'voice of the customer'. Within QFD both qualitative and quantitative data are collected using a variety of situated and non-situated techniques including direct observation, one-on-one interviews, focus groups and contextual inquiry. The results of this data collection are displayed using various graphical techniques including affinity diagrams (see Figure 3), tree diagrams and prioritisation matrices (Cohen, 1995). The affinity diagram shown in Figure 3 represents discussions with teachers and pupils regarding their needs for a particular information system, a PC based 'soft' keyboard. The diagram captures both verbatim statements of their desires, and also structures them along hierarchical lines, where compatibility relations and harmony can be represented, as shown in the diagram. These data collection techniques and graphical representations appear to be compatible with the early, analytic stages of information systems development and can be conducted in parallel, or alongside.

Figure 3 Affinity Diagram for Keyboard Users



Building and Using the House of Quality

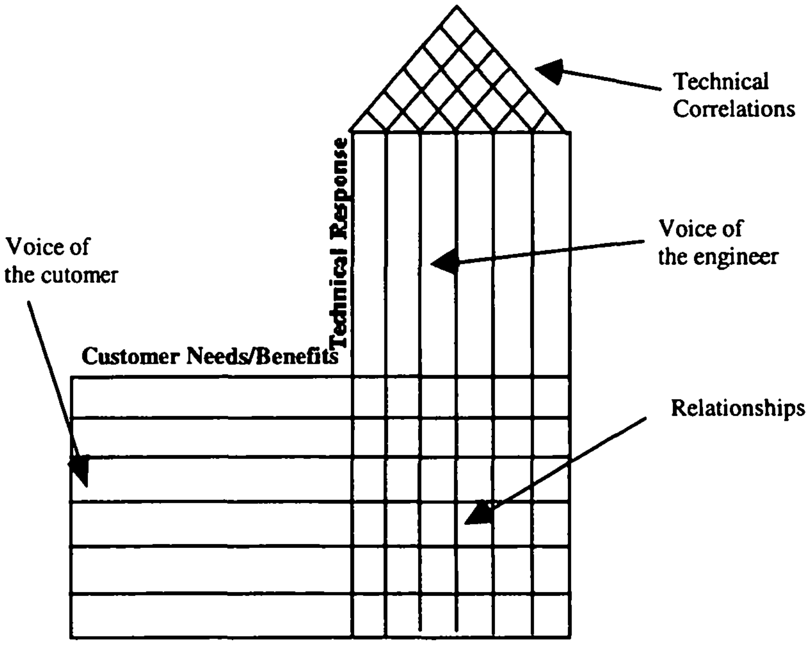
Figure 4 illustrates the most basic version of the HoQ. Many different extensions to the HoQ exist according to specific needs and contexts (see Cohen, 1995 and Akao, 1990 for a detailed review of the variations).

The output of the previous step is typically a hierarchical representation of the customer's needs, similar to Figure 3. The first step in building the HoQ is typically to list the customer's needs and benefits. This is then mapped against the technical response to those needs and benefits. Typically, the customer's needs and benefits are represented in the 'voice of the customer' (verbatims taken from interview notes or video taped observations) whilst the technical response is the engineer's or development organisation's response to the customer's needs, and is written in appropriately technical language. The technical response can be made at a number of different levels of abstraction, e.g. acceptance criteria, requirements or product features. If formal measurements are available for either the customer needs (from previous information system benchmarking for example) or the technical response, then these can be included in the HoQ along side their narrative descriptions. The HoQ then can be seen as a graphical tool for evaluating the relationship between the concerns of the customer and those of the developer.

The next step in completing the HoQ is to examine the relationships between these concerns. For each cell in the matrix, the cross-functional team (importantly, consisting of both customers and developers) enters a 'value' that captures the extent to which the engineering variables represented in the technical response contribute to overall customer satisfaction. This is perhaps the most important outcome of the HoQ analysis as it allows information system developers to prioritise their activities in line with customer needs.

The final step in completing the HoQ shown in Figure 4 is to complete the Technical Correlations matrix. This matrix captures the way in which different technical variables support or compete with each other. This information serves two main purposes. Firstly, by identifying which technical variables are interrelated it allows the engineers to identify key design bottlenecks. Secondly, if two technical variables are owned by different members of the cross-functional design teams then the correlations suggest communication pathways within the teams. Sometimes this is indicated in the HoQ by an arrow pointing from one technical variable to another that will be impacted if problems arise during its development.

Figure 4 The House of Quality



INFORMATION SYSTEMS AND THE HOUSE OF QUALITY

How would the use of the HoQ have helped in the examples discussed earlier where information systems ceilings were encountered?

Figure 5 illustrates the tension between providing blind people with headphones, thus improving the quality of auditory feedback they receive whilst isolating them from their surroundings. Clearly, a full HoQ analysis of this problem would contain many information system's requirements and technical responses, and many consequent correlations. For the purposes of illustration, only the tension introduced above has been represented.

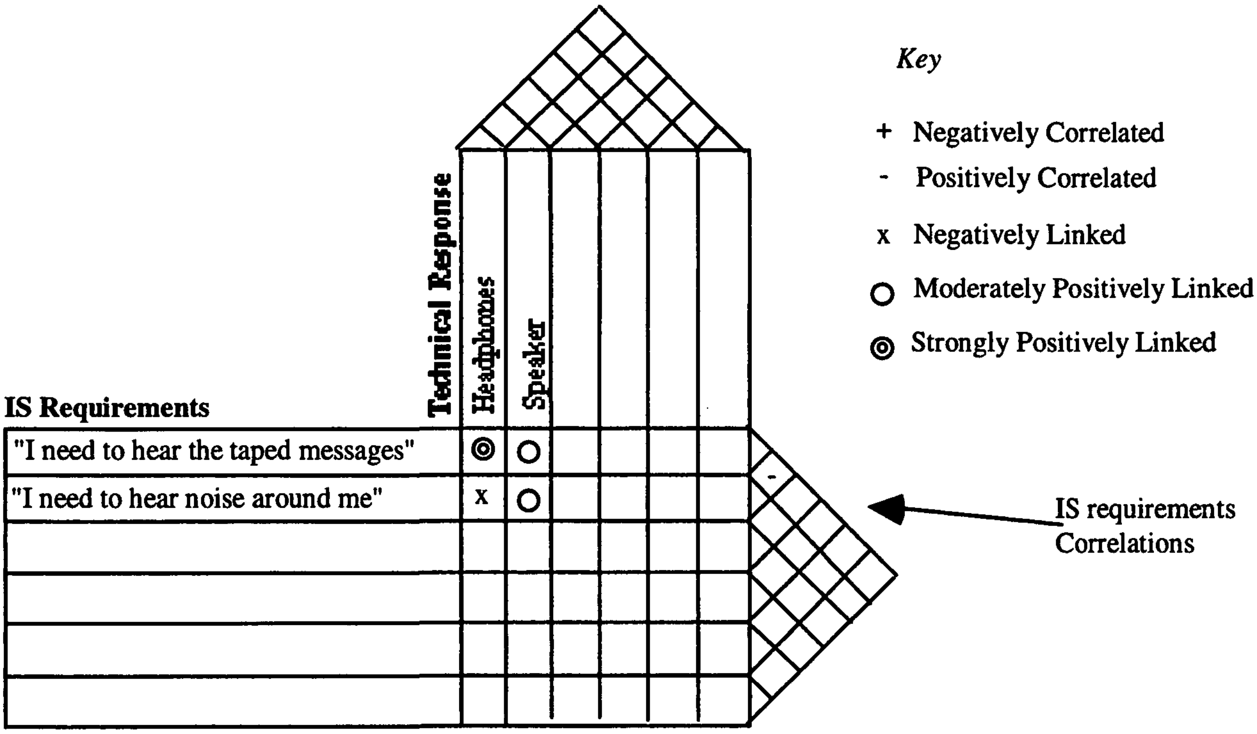
Figure 5 illustrates the user's concern (i.e. to be able to hear both the taped messages and the environment) and the engineer's response (the engineering options considered are to provide either headphones or an external speaker). From a technical perspective, the two engineering options are not seen as related (hence no symbol appears in the technical correlations 'roof' of the HoQ). Later in the analysis it may be that costs force these two options to become exclusive (and hence 'negatively linked') in the sense that both options cannot be provided within the budget for the product.

The relationships matrix indicates that, whilst providing headphones is strongly positively linked to the need for a clear audio signal, it is also negatively linked to the need to hear the surrounding audio environment. A compromise solution (from those enumerated) would appear to be the speaker that is moderately positively linked to both user needs.

The HoQ has been extended from that presented earlier to include an 'information system's requirements correlations' lattice. Just as technical correlations captured situations where technical factors were in tension, so the information system's requirements correlations lattice captures the ceilings, or tensions, introduced earlier.

Information system's requirements correlations inform the design team that the user has competing needs. Either compromises will be required on the part of the user, or a radical rethink of the design is needed. In this case it may be that the design team needs to consider providing blind people with a single headphone. We have found that it is sufficient to simply note that an information system's requirements correlation exists (by placing a '+' or '-' symbol in the appropriate cell) though clearly distinctions between different types of information system's requirements correlation are possible at this early stage, e.g. whether the requirements in tension are 'mandatory', 'nice-to-have' or 'safety critical'.

Figure 5 The House of Quality and Information Systems Ceilings


Technical Response
Headphones
Speaker
IS Requirements
"I need to hear the taped messages"
◎
O
"I need to hear noise around me"
x
O
-
Key
+ Negatively Correlated
- Positively Correlated
x Negatively Linked
O Moderately Positively Linked
◎ Strongly Positively Linked
IS requirements Correlations

An additional benefit of capturing information system's requirements correlations comes when the design is being conducted for a diverse stakeholder population. In this circumstance, information system's requirements correlations can span different stakeholder communities and their presence suggests the necessity for these communities to communicate with respect to their interrelated needs. This was the case with the expert/novice example discussed above. The information system's requirements correlations for this example would have highlighted that there was a tension between providing both groups with appropriate dialogue, feature access etc and that, if the tension could not be removed through design, the groups would need to be brought together and compromises would be required.

The HoQ then has different uses for different participants in information systems development. Developers can use it to summarise basic technical data and highlight the interrelations between technical variables. If the software being developed is to be a commercial product, then for marketing personnel the HoQ summarises the user's priorities whereas for management new product opportunities can be uncovered. Fundamentally, the HoQ encourages all of the stakeholder groups to understand and debate each other's priorities (Hauser and Clausing, 1988).

USING THE HOUSE OF QUALITY

Figure 6 shows an expanded HoQ modelling a problem in more detail. The design team were at the initial stages of designing a musical keyboard to be shipped with a simple animation package to be used by school children and information technology teachers alike. The keyboard was to appear on a computer VDU and notes were to be played, in sequence, using a mouse. Chords were not required. The animation package had a 'record and play' function that allowed strings of notes played on the keyboard to be recorded and played back at defined points in an animation, thus allowing students and teachers to add simple soundtracks to cartoon like animations.

The main challenge faced by the designers was that of satisfying both the teachers and the children with one design; an explicit constraint placed on the project by the customer. The HoQ in Figure 6 has been framed specifically around resolving this tension and so is rather focussed in what it models.

As shown in Figure 6, the teachers' primary needs were for a keyboard that allowed them to record the correct string of notes quickly, and that the keyboard should be of a conventional design. According to Fitt's law (Card, Moran and Newell, 1983), time to point to an object (using a mouse for example) is a function of the size of the object and the distance to that object and hence pointing time can, in part, be reduced by designing musical keys positioned closely together. Pointing time can be further reduced by maximising the control/display gain. However, both the pupil's concerns (to play strings of notes accurately) and characteristics (novice mouse users) suggested that accuracy be the preferred design criterion and so the designers should aim for large keys positioned far apart. This can be read in the technical response section of Figure 6.

The HoQ, as shown in Figure 6, was used as the basis of an heuristic walkthrough with both the users and designers present. The walkthrough was conducted on the design teams first prototype, the linear keyboard illustrated in Figure 7. It was clear from developing the HoQ that the two user groups had competing needs. Examining the technical correlations roof of Figure 6 indicated that the technical responses made to the teachers needs emphasised speed of selection whereas the technical response to the pupil's needs emphasised accuracy of selection. Furthermore, the teacher's need for a conservative keyboard layout conflicted with the teacher's need to play notes quickly; a conventional linear keyboard would result in some keys being far apart in space and consequently longer selection times. The linear keyboard of Figure 7 then appeared to suit neither the teacher's need for speed nor the pupil's need for accuracy of key selection. It did however satisfy the teacher's concern for a keyboard that conformed to expectations.

The second prototype produced the pie design in Figure 7. Here the designers produced a keyboard on which pupils could have large keys placed far apart (by playing around the periphery of the keyboard) and teachers could have small keys positioned close together (by playing in the central region of the keyboard). However, the pie keyboard clearly violated the teacher's requirement that the keyboard "conform to expectations".

Referring to the relative importance 'funnel' of Figure 6, previously the teachers had indicated that the IR requirement "conform to expectations" was more important than being able to "play notes quickly". The design team therefore decided to relax the design feature that minimised speed of selection (placing keys close together) in favour of a more conventional keyboard design. The resulting design can be seen as the hybrid prototype in Figure 7. This design allows the pupils to play around the upper region of the keyboard and the teachers to play around the more compact lower region. Further, the design adequately conforms to expectations (in the view of the users!).

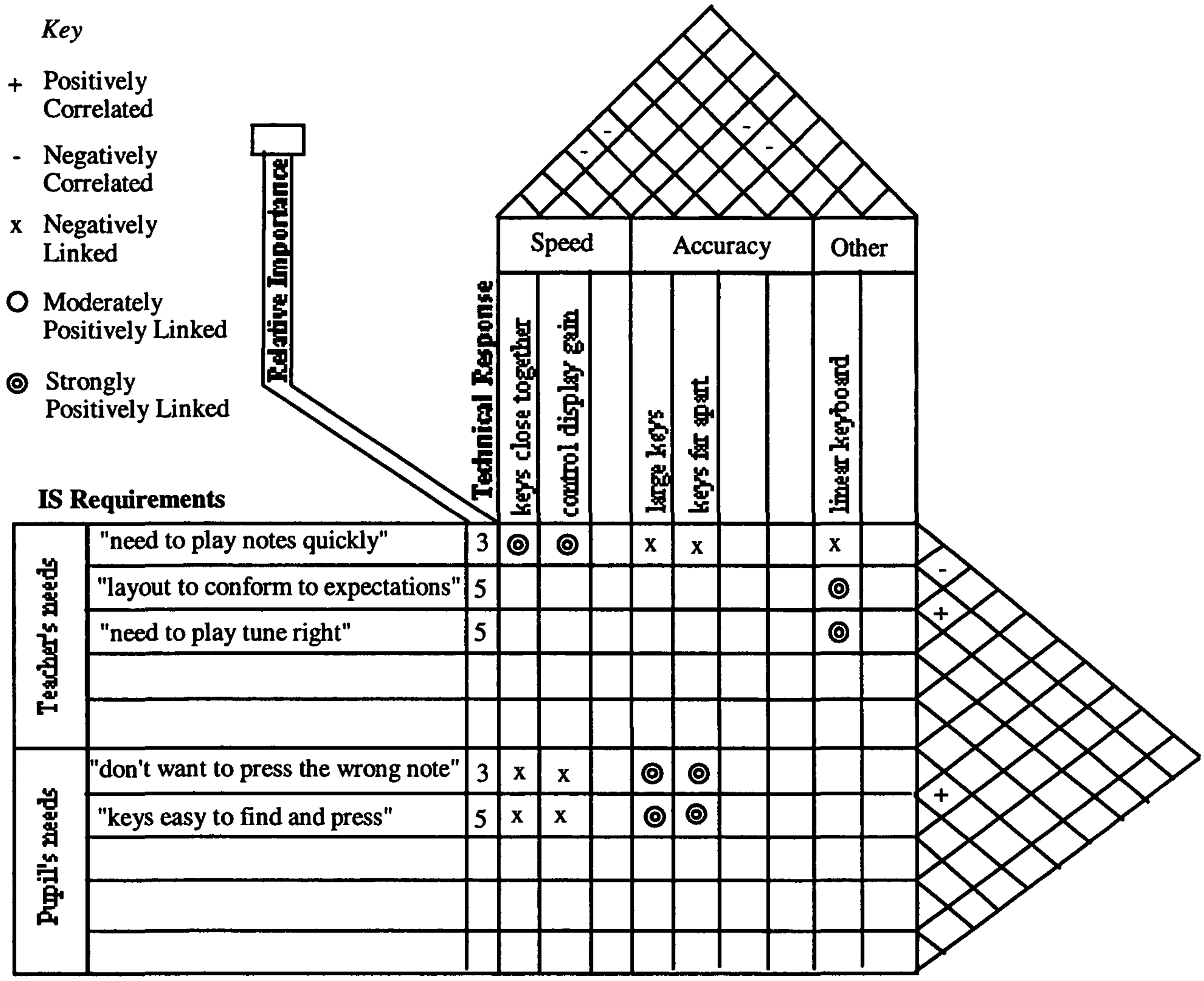
THE DIMENSIONS OF THE HOUSE OF QUALITY

The HoQ is the product of the assembly and interaction of the users, information systems, developers, development processes, organisational contexts etc. When using the HoQ to model information system requirements three aspects of both the information system requirements and the technical response become relevant:

Scope

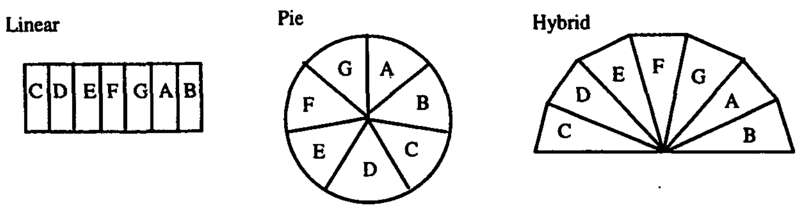
The scope of the information captured in the HoQ relates to the number of elements (i.e. instances of user, machine, context) represented. Ideally, the scope in the HoQ should be broad enough to include all significant elements in the intended information system, but not so broad as to be unfocused. Additionally, as the subsystem/cost tension illustrated, the HoQ may also include aspects of the development process itself, for example, the designer's characteristics, knowledge of development budget and project timeline. In principle, the scope may be very limited, and consist of an interaction between only two elements, or very extensive, and consist of many elements and interactions. We have found it best however to focus the development of the HoQ on a specific prioritisation problem, rather than attempt to model the entire design.

Figure 6 The House of Quality and Resolving the Ceiling



	IS Requirements	Relative Importance	Speed			Accuracy				Other	
			keys close together	control display gain		large keys	keys far apart			linear keyboard	
Teacher's needs	"need to play notes quickly"	3	◎	◎		x	x			x	
	"layout to conform to expectations"	5								◎	
	"need to play tune right"	5								◎	
Pupil's needs	"don't want to press the wrong note"	3	x	x		◎	◎				
	"keys easy to find and press"	5	x	x		◎	◎				

Figure 7 Prototype Musical Keyboards



Granularity

The granularity of the elements relates to the level at which the elements in the HoQ are specified or understood. The level of granularity is likely to be affected by many things, e.g. sophistication of the knowledge held by the agent of the elements concerned. For example, when a designer views all users as falling into the abstract class 'user' the designer is drawing on an abstract, or high-grain, view of the user. We have found it best to tailor granularity to need. The affinity diagrams produced prior to building the HoQ represent information system requirements in a hierarchical form, thereby potentially distinguishing between harmony and compatibility, and facilitate HoQ's being built at different levels of abstraction.

Balance

The degree of balance relates to the extent to which each element in the HoQ is able to contribute to the functioning of the overall system. Ideally, the balance in the HoQ should be such that priority is given to critical elements. Hence, to reiterate, achieving overall harmony may necessitate the sacrificing of certain compatibility relations. The aim of the HoQ is to make explicit the prioritisation process, and to record the sacrifices being made.

EXTENSIONS TO THE HOUSE OF QUALITY

Deriving the Dimensions

A variety of techniques are available for eliciting and structuring information system requirements and technical responses. QFD's recommendations include (Cohen, 1995) one-on-one interviews, contextual inquiry and focus groups with data from these sources being structured using affinity diagrams, tree diagrams and prioritisation matrices. Additionally, practitioners can employ task analysis to identify user problems and concerns. Alternatively, the Rich Picture technique from the Soft Systems Methodology (Checkland, 1981) can be used to elicit issues that are of concern to user groups, these can then be structured using affinity diagrams.

Metrication

The symbology used to represent the mappings and correlations between the requirements and the responses can be expanded in a variety of ways. When available, benchmarks and criteria can be used to metricate the information system requirements. Work in software quality measurement (see Akao, 1990, chapt 12) is working towards metrics useful for formalising the technical response. A large number of different symbol systems is used for indicating mappings and correlations, ranging from the graphical notation presented above, through to statistical and mathematical treatments. This will be an ongoing area of work as QFD and HoQ are used in information systems development. We have found the graphical notation presented in Figure 6 to be satisfactory for supporting the resolution of information systems tensions.

Modelling Actual-Present and Desired-Future

Akao (1990) presents a HoQ expanded to model temporal aspects of the development process. Essentially, a HoQ is produced that models the actual-present and desired-future situations. In this way ongoing design is given a firm reference point of current problems and concerns. Further extensions include calender-dates that relate the future products to present problems and development processes.

The Next Step

In manufacturing applications of QFD a series of HoQ's are produced (named 'streets'), each feeding into the next. It is possible to conceive of steps in the information systems development process being framed by the drawing of a HoQ, and thereby encourage the flow of customer priorities right through the development process. As illustrated in Figure 5 above, the first HoQ would map information system requirements (the rows or 'whats') onto high level technical concerns (the columns or 'hows'). Next the high level technical concerns (e.g. 'headphone') become 'whats' to the new 'hows' of low level solutions (i.e. detailed technical specifications of specific headphone sets). Eventually an attempt could be made to map the final product back onto the original requirements.

These linked houses would therefore convey the user's voice through the information systems development process. This process would facilitate product evaluation and traceability. Additionally, in representing the temporal aspects of the design process the HoQ's produced capture the unfolding design rationale.

Streets are therefore one way of overcoming the limitation on the complexity of what is modeled that is imposed by the simple 2 dimensional matrix (particularly limitations in the scope of the requirements and the granularity of their representation- see above). Streets can be constructed that stretch across the development life cycle (as discussed) or hierarchically arranged and thus supporting the refinement of requirements at lower and lower levels of abstraction.

CONCLUSION

This paper has argued that information systems design needs to concern itself not just with 'how' to engineer compatibility between all the elements of the information system, but also, given that relations often compete, 'which' compatibilities to give priority to. The HoQ was introduced as a mechanism that cross-functional design teams could use to represent competing information system requirements and explore the relationships between the 'voice of the user' and the 'voice of the developer'. We use the HoQ as both an investigation tool and also an analysis approach. We have found that completing the matrix imposes a rigor on the discussion of requirements that adds value to the kinds of collaborative cross-functional sessions we are conducting. We are using the notation throughout the lifecycle and constantly refining our understanding of the user's needs in the light of our growing enlightenment about what is and is not possible, what is and is not wise. We have used it as both a stand-alone technique and in conjunction with scenario based design and the soft systems methodology (Howard and Dearden, 1998, Dearden and Howard, 1998). In each case it acts to clarify the relations between the concerns of multiple stakeholder groups and assists us in prioritizing the information systems requirements. In order to manage the complexity of data that can result, software tools are being developed by the author and others (see for example QFD2000 and QFDCapture).

The HoQ adds to the growing body of participative design techniques being used in Information Systems (eg. Checkland, 1981, Avison and Wood-Harper, 1990). Extensions specific to the use of the HoQ for modelling information system tensions were discussed.

How does the HoQ respond to three aspects of wicked problems introduced earlier?

- *The lack of a definitive formulation:* The HoQ is used by multi-function design teams to come to agreement on both the relationships between their concerns (e.g, users and designers) and the prioritisation of those concerns. In that sense the HoQ can be seen as the team's operationalisation of the design qualities under discussion, including information system requirements. Although arriving at a definitive definition of the requirements is beyond the notation, the HoQ does at least allow the current working definition of the information system requirements to be articulated and debated.

- *Solutions to information system problems tend to be more or less satisfactory:* The HoQ directly supports the team in prioritising the customer attributes and technical response. In that sense, both the nature and the degree of the solution's 'satisfactoriness' is articulated and recorded.

- *The solution space to any information system problem is not exhaustively identifiable:* In recording the nformation systems requirements and the technical dimensions of the associated response, the HoQ captures the limits of the search conducted during the development of the information system. As with design rationale techniques, the house both encourages a more considered search and captures the extent of the searched space (and by implication, options not considered).

REFERENCES

Akao, R. 1990, **Quality Function Deployment**. Productivity Press, Cambridge, Mass.

Al-Siuru, P. 1993, Using QFD Techniques in User Interface Specification: first experiences. In **Proceedings of the Fifth International Conference on Human Computer Interaction**. Pub: Elsevier Science.

Avison, D. and Fitzgerald, G. 1988 **Information Systems Development: Methodologies, Techniques and Tools.** Oxford: Blackwell Scientific Publishers.

Avison, D. and Wood-Harper, T. 1990 **Multiview Methodology**. Oxford: Blackwell Scientific Publishers.

Bias, R.G. and Mayhew,D.J. 1994, **Cost-justifying your usability**. Boston, Academic Press.

Card, S., Moran, T. and Newell, A. 1983, **The Psychology of Human Computer Interaction**. Pub: Elsevier Science.

Checkland, P. 1981, **Systems thinking, systems practice**. Chichester: John Wiley and Sons.

Churchman, C.W. 1978, Philosophical Speculations on Systems Design, **Handbook of Operational Research** J.Moder, S. Elmaghraby (Eds), Van Nostrand Reinhold

Churchman, C.W., 1971, **The design of inquiring systems: Basic concepts of systems and organization,** Basic Books Inc Publishers.

Cohen, L. 1995, **Quality Function Deployment: How to make QFD work for you**. Addison Wesley.

Dearden, A. and Howard, S. 1998, Capturing user requirements and priorities for innovative interactive systems. In Proceedings of **OZCHI'98, Adelaide, Nov 1998.**

Howard, S. and Dearden, A., 1998, Innovating Usable Software: Integrating QFD with the Soft Systems Methodology. Published in the Proceedings of **World Innovation and Strategy Conference (WISC'98)**. 2-5 August, Sydney, Australia.

Hauser, J.R. and Clausing, D. 1988, The House of Quality. **Harvard Business Review**, May-June, 1988.

Lundell, J. and Williams, D. 1993, Integrating QFD into Software Development. In **Proceedings of the Fifth International Conference on Human Computer Interaction**. Pub: Elsevier Science.

Petrie, H., Johnson, V., Strothotte, T., Michel, R., Raab, A and Reichert, L. 1997, User centred design in the development of a navigational aid for blind travellers. In S.Howard, J.Hammond and G.Lindgaard (Eds) **Human Computer Interaction, INTERACT'97**. Chapman and Hall.

QFDCapture, 1999, WWW http://qfdcapture.com/webpage/ (accessed June 8, 1999).

QFD2000, 1999, WWW http://www.users.globalnet.co.uk/~tqs/ (accessed June 8, 1999)

Rittle, H.W.J. and Webber, M.M. 1973, Dilemmas in a general theory of planning. In **Policy Sciences**, 4, 155-169.

Shindo, H. 1991, **Quality Function Deployment Applied to Software at Digital Equipment Corporation.** Thayer School of Engineering, Dartmouth College, May 1991.

Stowell F. A. and West D., 1989, Expert systems: ramifications for the knowledge engineer, In **Systems Analysis, Modeling, Simulation** 16(9).

Stowell F. A. and West D., 1990, The contribution of systems ideas during the process of knowledge elicitation. In R.L.Flood, M.C.Jackson and P.Keys (eds.) **Systems Prospects - The Next Ten Years of Systems Research**,

Teasley, B. and Scholtz, J. 1997, User-centred GUI standards design: a case study. In S.Howard, J.Hammond and G.Lindgaard (Eds) **Human Computer Interaction, INTERACT'97**. Chapman and Hall.

Vicente, K.J., Burns, C.M. and Pawlak, W.S. 1997, Muddling through wicked design problems. In **Ergonomics in Design**, January, 1997.

Zultner, R. 1992, **Quality Function Deployment for Software**. American Programmer Magazine, February, 1992.