Engineering with Computers (2000) 16: 236–252
© 2000 Springer-Verlag London Limited

Engineering with
Computers

Improving the Rationale Capture Capability of QFD

Y. Reich

Department of Solid Mechanics, Materials and Systems, Tel Aviv University, Ramat Aviv, Israel

Abstract. *The goals of Design Rationale Capture (DRC) are improving design quality and reducing design time. These general goals have led to the design of many DRC techniques originating from research in various design and other related disciplines. However, little evidence about attaining these goals or about their usability or utility in practice has been demonstrated. To improve this situation, we use QFD tools to design a new DRC technique. QFD can be used to transform general needs into working products; its practical utility in improving design quality and reducing overall design time are widely recognised. In the course of using QFD, significant design knowledge is generated and recorded. Consequently, we picked QFD as a candidate DRC technique, and used QFD itself to improve its DRC properties. The utility of QFD as a DRC tool is illustrated by using it to capture the rationale underlying its own design.*

Keywords. Collaborative design; Communication; Design argumentation; Design practice; Design rationale; Graph models; House of quality; Quality function deployment; Research methodology; Shared memory

1. Introduction

Design Rationale (DR) is a record of the reasoning process and information that underlie a particular artifact design. DR may include various design decision, issues, alternative choices, the arguments for or against different choices, their evaluations and assumptions, as well as additional relevant information. The premises underlying DR research have been that *DR is useful and usable for designers*, and that in capturing it, design quality could be improved and its duration shortened [1]. More specifically, DR researchers have become convinced that the availability of DR can assist designers in future design scenarios, and even contribute *within* a long-term project whose DR is continuously recorded. The proponents of DR suggested that DR can assist in the following design activities [1–3]:

- A1: Structuring design problems or spaces.
- A2: Performing various computational services such as: the maintenance of design consistency or design verification.
- A3: Communicating reasoning and perspectives to others.
- A4: Documenting design.
- A5: Creating and accumulating design knowledge.
- A6: Debugging, modifying, redesigning or maintaining designs.
- A7: Critically reflecting during design.

Through these activities, and others, designers would be able to answer *how*, *why*, *by whom* and *when* a particular artifact was designed to satisfy its goals. They might be able to ask various 'what if . . .?' questions, and use previously captured rationales to short-cut the development process of new artifacts and improve their quality. To materialise this potential, DR research has explored a range of techniques between two extremes:

- E1: DRC for use primarily by computer programs.
- E2: DRC for use primarily by human users.

The first extreme involves creating formal languages that can be manipulated by computer programs. However, these approaches are restricted to well understood design problems. Therefore, most DR research has concentrated on the second extreme. It involves creating semi-formal (structured) languages that users can use to record various kinds of information related to design work, and that other users can read, interpret and reuse. These languages are sometimes called argumentative DR.

Correspondence and offprint requests to: Professor Y. Reich, Department of Solid Mechanics, Materials and Systems, Tel Aviv University, Ramat Aviv 69978, Israel.
E-mail: yoram@eng.tau.ac.il

Upon examining the DR literature, we find little evidence to support the aforementioned premises about the usefulness and usability of DRC [1]. This observation triggers a systematic analysis that ties the aspirations of DRC to the design of existing DRC techniques, and leads to the design of improved, useful and usable DRC techniques. These analysis and design tasks can be supported well by Quality Function Deployment (QFD).

QFD tools emerged from Japanese design practice as means to integrate customer requirements and other lifecycle concerns into product design for improving its quality [4,5]. The popularity and practical success of QFD is detailed in many recent publications [5–8]. To use QFD in these analysis and design tasks, we treat designers as the customers of DRC techniques, and the product of interest as a DRC technique. The application procedure involves an analysis of existing DRC techniques which, in turn, produces a record of the issues underlying DRC tool design. The proven practical success of QFD tools and their ability to capture rationale make them attractive candidate DRC techniques. We will discover that QFD tools provide solid baseline on which improved DRC techniques can be developed.

The remainder of this paper is organised as follows. Section 2 briefly reviews research on argumentative DR. Section 3 describes the use of QFD tools in analysis and design tasks in general, and in the design of a DRC technique specifically. The outcome of this section is a proposal for a DRC technique based on QFD and its underlying rationale. Section 4 demonstrates the reuse of DR captured by QFD tools in diverse ways including improving design quality and supporting new design situations. Section 5 concludes the paper.

2. Review of Argumentative DR Research

The central focus of DR research has been the development of mechanisms for recording design arguments. This was motivated by Toulmin's research on argument representation in the form of justifications [9]. However, Toulmin's approach is inappropriate for expressing many arguments related to design [3]: it is too general; its template can easily confuse users, leading them to argue about the template roles and not the relevant design issues; and it causes DR object types to be context dependent.

Another motivation and a source of inspiration for argumentative DR research has been the work on IBIS (Issue Based Information System) [10], and the development of computer tools that support the generation of IBIS-like models such as gIBIS [11]. As shown in Fig. 1, graphical IBIS models look like graphs of nodes and links, where nodes can be *issues*, *positions*, *arguments* and *resolution*, and links include *support*, *questions*, *replaces*, *generalises* and their opposites. Again, the ability of IBIS or gIBIS to represent DR is limited [3].

There have been many extensions to IBIS models for addressing various issues that developers saw as critical. One example, PHI, improved the model of the argumentation process by changing the meaning of what constitutes an issue and allowing to build hierarchies of issues [12]; nevertheless, PHI still, has many limitations [3,13]. Another extension to IBIS, Potts' and Bruns' [14] model shown in Fig. 2, included explicit representations of the artifact and design steps.

The motivation behind incorporating explicit representations of the artifact being designed into DR languages is that it would: (1) reduce the documentation time (since it captures what engineers do when they do it); (2) improve documentation quality (e.g. in ADD [15]); (3) focus the argumentation system on issues relevant to the design (e.g. in JANUS [13]); (4) provide better support for communication (e.g. in REMAP [16]); and (5) reduce negotiation time (e.g. in XNetwork [17]).

The motivation behind incorporating represen-

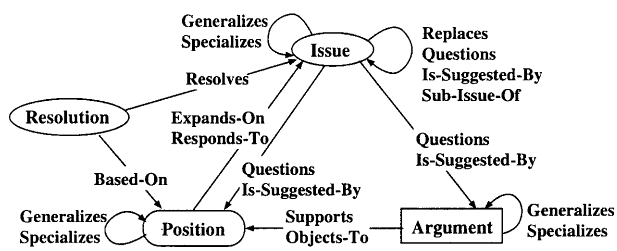


Fig. 1. A variant of gIBIS notation.

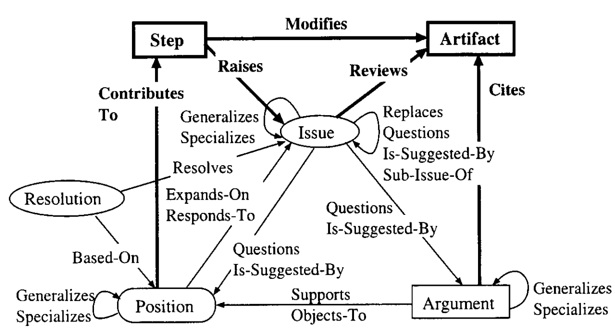


Fig. 2. Potts and Bruns notation.

tations of design processes into DR languages is that it would reduce the cognitive load of users, and would allow for capturing the temporal and causal dependencies between design decisions, constraints and the artifact evolving state. In particular, when the DR language is embedded within a process model developed after an empirical study of a particular task such as the REMAP model, there might be better chances to succeed in using the tool in practice. Beside REMAP, Potts and Bruns model, as well as JANUS, ADD and XNetwork integrate a process model with the use or capture of DR. The incorporation of a specific process model into the DR and using it to assist design is a need that raises an issue which needs to be addressed by any DRC technique (the issues are numbered in the order they appear in the text):

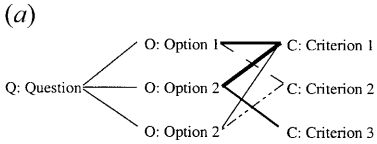
I1: How should a process model be incorporated in a DRC technique?

DR languages are usually perceived as fixed formalisms. However, Potts and Bruns argued that a DRC technique should be capable of been tailored to suite its intended domain. Such tailoring has the effect of reducing the cognitive gap between designer thoughts, domain requirements, and the formalism. In particular cases, such as in *m*SIBYL [18], JANUS or ADD, such tailoring with the addition of domain knowledge, improved the kinds of computational services that could be made available. Providing computational support for, and management of such tailoring activities is not trivial, although two systems, KMap [19] and more so, *n*-dim [20,21] address this issue which we define as:

I2: How do we design DRC techniques that support flexible formalisms?

IBIS, gIBIS, Potts and Bruns and several other argumentation notations are meant to be used *while* designing. Another approach, called QOC [22,23] (see Figs 3(a) and (b)), was developed to be used in retrospect, not for recording the design process, but for recording the *structure of the design space*. The idea underlying QOC is that spending additional recording effort provides broader context for future decision making even if some process details are lost. Similar to previous approaches, QOC has its own limitations [3].

(*a*)



(*b*)

Options	Criteria 1	2	3
1	5	2	-
2	5	-	4
3	3	1	-

Fig. 3. QOC notation.

There have been other approaches that integrated design process and design space information. SIBYL employs a representation language called DRL that implements extensions to gIBIS and some concepts from QOC. DRL supports several computational services on DR including [24]: (1) dependency management (between interdependent DR components); (2) plausibility management (by maintaining dependency between claims); (3) viewpoint management (using the dependency between DR components sharing certain assumptions); and (4) precedence management (by detecting dependency between decision sharing the same goals). SIBYL computational services are based on adding meaning to the roles of parts of the DRL language. These additions increase DRL's expressiveness, but make its use harder for humans.

Finally, there have been studies that sought to integrate the two extreme DR research approaches mentioned before, by focusing on formal languages, such as DRCS [25], that can be used by humans and software tools. DRCS was intended to be generic for design tasks, as opposed to languages specific to particular design tasks such as JANUS or REMAP, or approaches that are not directed solely at design such as gIBIS. However, DRCS is very elaborate, thus could become cumbersome to use.

Summaries of additional argumentative (and other types of) DRC techniques can be found elsewhere [1,3]. These summaries refer to techniques originated in design research, and some from other disciplines such as collaborative writing and student teaching. Lists of issues relevant to DRC techniques also appear elsewhere [26]. Most often, these issues are generated following analyses of needs from such systems, as well as analyses of existing tools. These sources should be consulted when developing new techniques.

None of the systems mentioned in this section, as well as others published to-date, have enjoyed practical success. Moreover, Ullman's [26] 13th issue is about the testing of DRC systems, and whether they could be assessed without building them. We take a position on this issue and interpret practical success and its assessment not as a proof of concept in experimental research setting [27,28] but as a wide acceptance in, and use by, industry. (More on the issue of research and practice can be found elsewhere [29]). Consequently, we set our general task to improve upon this situation, and develop a method with better potential of being practically successful.

Unfortunately, the description in this section is too general to serve as a basis for appreciating the issues involved in DRC, their interdependencies, and relative importance. It is, however, representative of the kind of textual analysis that can be found in the DR literature. To better analyse DRC techniques and provide sufficient ground for designing new techniques, we employ tools from a branch of quality product design called quality function deployment. Using these tools, we collect additional design issues and address them.

3. Designing with QFD Tools

QFD is a constantly evolving set of tools aimed at improving product design quality. While some associate QFD with the House of Quality tool, it is indeed much more comprehensive. Different perspectives include in QFD the HoQ, the seven new management tools [30], Pugh's concept selection technique [31], and many other tools. This evolution can be appreciated by studying the published papers, and those presented at the annual symposiums on quality function deployment [8].

QFD tools are used to elicit information from various sources, including customers, engineers and past product performance, and organise them in various ways so that they can be reused for planning and design. QFD tools represent information in simple graphical models that are easy to comprehend. There are several stages involved in using QFD. In the *information acquisition* stage, the literature is reviewed and a conceptual model of the domain is generated with several QFD tools. As design progresses, the focus shifts to *information management*, and finally, to *information use*. *Information accumulation and communication* is done by saving QFD diagrams in easily accessible repositories. This section reviews QFD tools, and discusses their application in design information acquisition, management and use.

3.1. Seven Management Tools

The seven management tools were developed in the 1970s as means to improve design quality; they include [30]:

T1: *KJ method* (*affinity diagram*), which is a brainstorming tool, clarifies important concepts by collecting and organising diverse verbal data (e.g. ideas, concepts, issues, opinions, etc.) into groups. Its results can serve as common terminology to be used by project participants, thereby improving the future communication of design information. Future designers would be able to better understand the meaning of terms by tracing the ways in which terms were proposed, grouped and constructed. Affinity diagrams provide the basis for supporting activities A3 and A5.

We did not use affinity diagrams in the design of DRC techniques, mainly because they are best suited for group design, while the present design was conducted by one person. We used the other tools which are useful for single as well as group design.

T2: *(Inter) Relation diagram* clarifies the logical relationships between concepts related to an artifact. For example, it can record issues, alternatives, criteria for selecting them, other DR information, as well as other concepts that are seldom or never captured in DR languages.

We have used relation diagrams to record concepts related to DR while studying the DR literature. Figures 4–6 record each concepts and

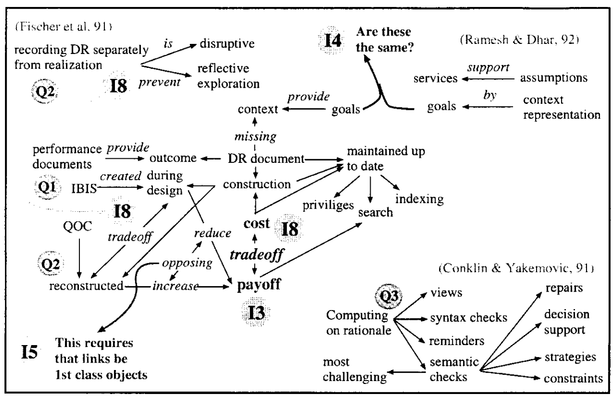


Fig. 4. Relation diagram: DR issues, mainly from Conklin and Yakemovic [2].

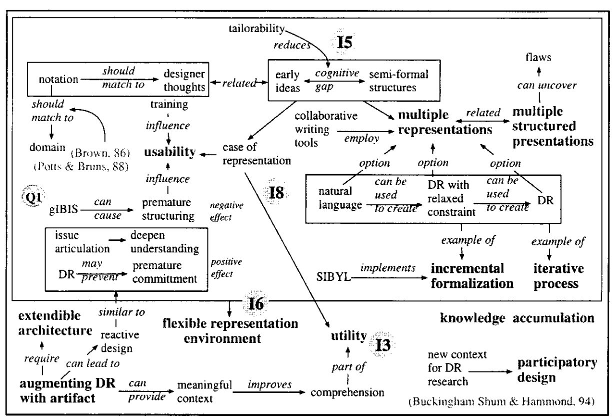


Fig. 5. Relation diagram: DR issues from Buckingham Shum and Hammond [1].

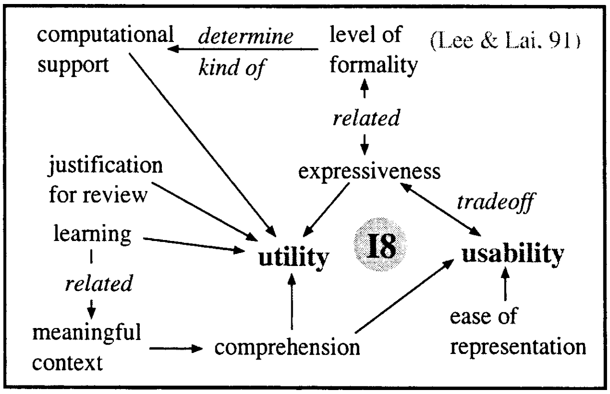


Fig. 6. Relation diagram: DR issues from Lee and Lai [3].

relationships about DRC from a single reference. The diagrams were created while analysing the text. They are quite overloaded because they each summarise several separate diagrams. Also, their reading is made harder since the printed version is black and white. While examining the diagrams that have emerged, we observed issues related to argumentative DR languages and labelled them on the diagrams with I*i*, where *i* denotes the issue number. The issues are:

- **I3**: How is the meaning of terms such as usability or payoff captured for future users? This is a critical issue. Some of the terms in the figures are not trivial to understand, and their interpretation is subjective given the little contextual information in the figures. Improving the capturing of meaning involves using the affinity diagram in the first stage as well as providing links from the diagrams to the source material.
- **I4**: How do we know whether two terms with the same label extracted from different DR studies mean the same? The solution of this issue depends on the solution of issue **I3**.
- **I5**: How could statements about, or references to, links be represented?
- **I6**: How could complex relations between complex sub-diagrams and a conclusion originating from them be represented?
- **I7**: How could complex relational arguments be structured?
- **I8**: How are diagrams detailing similar issues recognised to be similar (e.g. the clusters of concepts around the utility-usability and cost-payoff trade-offs)?

There is another critical issue that arises from the flexibility of relation diagrams. Clearly, they can record any structured information. Hence, their effectiveness depends on judicious use based on experience and discipline. Supporting the accumulation of such experience is an important issue which is handled by addressing issue **I2**. The effectiveness of relation diagrams (and other tools) can be enhanced by developing mechanisms for detecting contradictions or other flaws in relation diagram models, leading to the following issue:

- **I9**: Which mechanisms could be developed for detecting contradictions or other flaws in relation diagrams?

Relation diagrams are quite informal, a good property for the initial stages of problem understanding or information acquisition. Subsequently, their data can be further structured in other QFD tools. Figure 7 shows an intermediate diagram created from several DR studies in order to understand the incorporation of artifact representation into argumentative DR languages.

In summary, relation diagrams mainly support activities A1, A3, A5 and A7.

T3: *Tree (systematic) diagram* is used to decompose a concept into its contributing factors; for example, the concept might be a problem or a plan, and the factors might be solution steps or components, respectively. When analysing and organising data recorded in relation diagrams, tree diagrams of goals, issues, alternatives, criteria, assumptions, tasks and other DR aspects may emerge as shown in Figs 8 and 9.

Figure 8 records the influence of the seven design activities mentioned in the introduction on information quality or product quality, and on

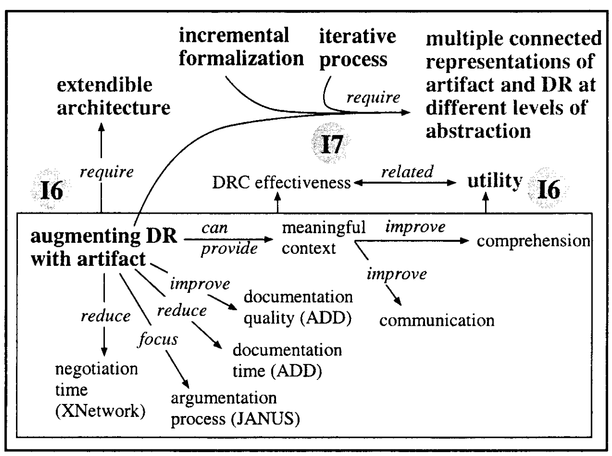


Fig. 7. Relation diagram: Integrating DR with artifact representation.

design duration through three mediating factors that characterise different design stages. The influences are qualitative. Subsequently, some of the relationships in this and other diagrams are exported to other QFD tools, where they are given additional qualitative and quantitative interpretations.

Tree diagrams can support simple computational services to answer questions such as 'is the top goal satisfied?' For example, if all the four prerequisites to achieving product quality in Fig. 9 are met, the term *achieving product quality* could be flagged as satisfied.

The use of tree diagrams in different domains might lead to evolving different languages with different typed terms and relations. It would be highly beneficial for users to have a facility that allows for easily evolving such languages. Designing such a facility was referred to before as issue **I2**.

Tree diagrams can support activities A1, A2, A3, A4, A5 and A7.

T4: *Matrix diagram* clarifies the relationships between different facets of an artifact such as functions, tasks or requirements, and identifies their relative importance. There are several types of matrices, depending on the number of facets that are used. An L-shaped matrix involves two facets; a C-shaped matrix involves three facets in a 3D space, and is therefore hard to conceptualise or construct on a piece of paper; and an X-shaped matrix involves four facets. Two important types of matrix techniques are the HoQ and Pugh's concept selection. They are further explained in Sections 3.2 and 3.4, respectively.

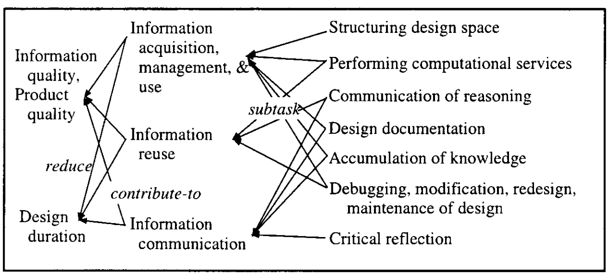


Fig. 8. Tree diagram: Contributions to information and design quality.

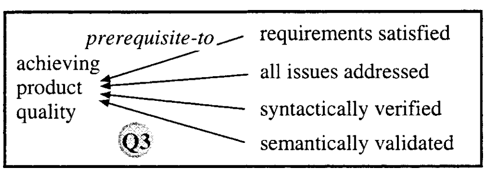


Fig. 9. Tree diagram: Prerequisites to achieving product quality.

Matrix diagrams can express the relations between several DR components. To illustrate, alternatives, criteria and arguments can become the three axes of a 3D matrix such that each *ijk* cell in the matrix contains the evaluation of the *i*th alternative with respect to the *j*th criteria, due to the *k*th argument. However, 3D matrices are not easy to manipulate with paper methods. Therefore, another issue is:

I10: How can we provide usable support for manipulating and visualizing 3D matrix data?

Moreover, we may wish to enter evaluations that are more complex than numbers. This leads to the next issue:

I11: How can we support evaluations that are more complex than numbers?

Matrix diagrams support all the seven design activities.

T5: *Matrix data analysis* include techniques that provide a compact perspective of the data in matrix diagrams. The techniques include: displaying matrix diagram data in a graphical form, most often generated by principal components analysis or clustering; executing statistical methods; or performing any other computation that can be defined on matrix data. As a result of these analyses, interesting and occasionally surprising or insightful concepts and relations can emerge and assist in critical reflection. These concepts can occasionally be considered as new domain knowledge. Thus, matrix data analysis supports activities A2, A5 and A7.

For the sake of completeness we describe the two remaining tools, and return to describing the use of the HoQ, Pugh concept selection, and matrix data analysis in the present design, in the next section:

T6: *Process Decision Program Chart (PDPC)* is used to uncover and display all the events (expected or undesired) that might happen when implementing a plan. Events are listed with their possible outcomes and possible counter-measures if required. This tool can borrow input from a tree diagram that details plans and proceed with the use of Failure Mode and Effect Analysis (FMEA). The use of PDPC forces designers to critically reflect upon their design, locate possible weaknesses, and identify remedies that may require redesign. PDPC serves as a record of the reasons underlying such redesign. In addition, PDPC expands the design space developed using other tools.

PDPC supports activities A1, A3, A4, A6 and A7.

T7: *Arrow diagram* is used to schedule or select the most appropriate plan. It is similar to a Gantt chart, and can be used for critical path analysis and for other project planning purposes. An arrow diagram records the reasons for selecting a particular implementation procedure. It can be used to answer questions such as: 'what will be the result if we select plan X instead of Y?'

The arrow diagram supports activities A2, A3 and A7.

Thus far, the focus of designing the new DRC technique has centred around information acquisition and its organisation. This is the basis of pursuing design with the more complex tools.

3.2. The House of Quality

The HoQ is a central QFD tool. Its use involves several steps, each of which is associated with filling a room in the house (rooms are marked by S*i* in Figs 10 and 11); for further details on using the HoQ consult Akao [4], Clausing [6] and King [5]:

S1: *Extracting customers' voice.* This step extracts users needs or requirements of the design. The needs can be design issues or goals, a product structure, or other items that need to be attained or implemented.

The needs in Fig. 10 were formulated from DR research and empirical studies of design studied through the use of the relation and tree diagrams discussed before:

N1: Reduce design time
N2: Improve product quality
N3: Prevent loss of expertise (retirement, early leave)
N4: Improve communication with customers
N5: Support company growth
N6: Be inexpensive

S2: *Performing competitive analysis of designs with respect to customers' voice*. This step rates different designs with respect to the requirements extracted in the first step; it is similar to using a QOC matrix (Fig. 3(b)).

In Fig. 10, Step S2 records a comparison between existing DRC techniques. Note that, for the purpose of using the HoQ, the evaluation of the different techniques need not be precise or perfectly accurate. Imprecision may be introduced, since each technique is evaluated separately, and not necessarily compared to other techniques. Repertory grid tools such as KSSO [32], as well as various matrix data analyses techniques, can assist in or provide feedback on this process. This topic is elaborated further in Section 3.3.

An important product of this room is the normalised weight of customer requirements calculated as follows:

$$r_i = t_i/c_i \tag{1}$$

$$w_i = r_i s_i i_i \tag{2}$$

$$wn_i = w_i/\sum_i^N w_i \tag{3}$$

where c_i is the evaluation of the baseline design with respect to the *i*th customer requirement, t_i is the target value of this requirement, set by the designer (or design team); i_i is the importance of this requirement, set by customers; and s_i is an evaluation of the 'selling power' of this requirement, set by designers. In Fig. 10, QFD was selected as the baseline design, and the most important customer requirements turned out to be *improving design quality, preventing loss of expertise* and *reducing design time*.

S3: *Expressing the engineer's voice*. This step involves defining product attributes or technical terms that contribute to satisfying customers' requirements. These attributes can be positions about how to address issues as in IBIS, or even options that address questions as in QOC.

Figure 10 depicts the attributes imported from the tree diagram in Fig. 9.

S4: *Uncovering correlations.* This step identifies the contribution of the technical terms towards satisfying customers' requirements. The correlations can record the 'strength of contribution' of positions to addressing issues which is more expressive than the corresponding binary relation possible in IBIS-like languages (i.e. supports and objects-to). Also, the correlations can represent the strength of optional answers which is similar to the relations in QOC.

It is required that no line in room 4 remains

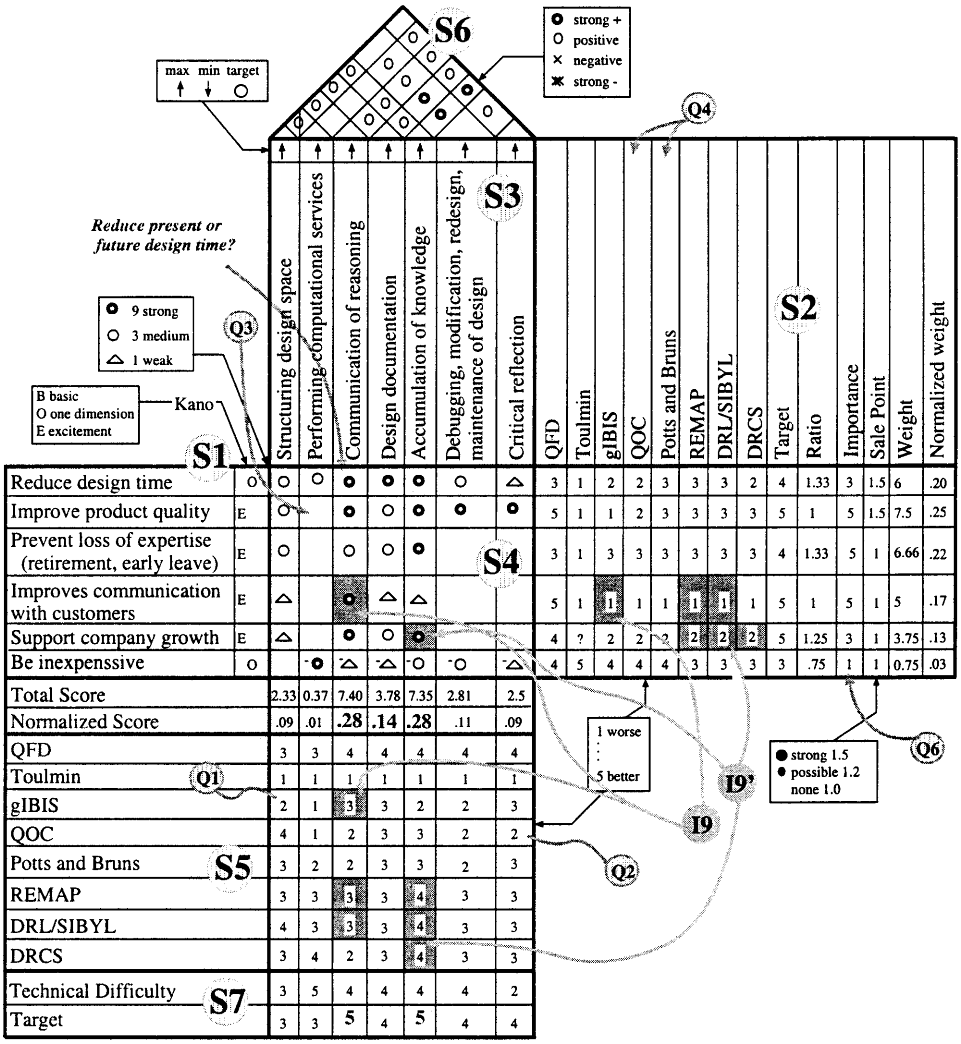


Fig. 10. HoQ: mapping DR requirements to DR tasks.

empty, making the customer requirement on that line unattainable. Equally, it is not best if the matrix is full, thus displaying too many interdependencies between the technical terms (i.e. the design is highly coupled [33]). Some of the entries in Fig. 10 were previously detailed qualitatively in relation or tree diagrams (e.g. in Fig. 8). When they were quantified, some obtained high scores and others were eliminated. Also, the last line of room 4 contains negative values, which are not permitted in regular HoQs but are introduced here.

S5: *Performing technical comparison and calculating scores.* This step evaluates existing designs according to the technical terms and calculates the normalised score of the technical terms as follows:

$$ti_j = \sum_{i}^{N} wn_i h_{ij} \tag{4}$$

$$tin_j = ti_j / \sum_{j}^{M} ti_j \tag{5}$$

where h_{ij} is the influence of the jth technical term on attaining the ith customer requirement.

In Fig. 10, the technical comparison constitutes a multi-criteria evaluation of the DRC techniques we compared in Step S2. We used KSSO to assist us in this comparison (see Section 3.3). The most important technical terms for achieving the requirements of DRC techniques are the *communication of reasoning* and the *accumulation of knowledge*.

We note that, thus far, there have been three independent evaluations in rooms 2, 4 and 5, but they are really related. For example, I9 in Fig. 10 points to evaluations that seem mismatched. If *communication of reasoning* strongly influences

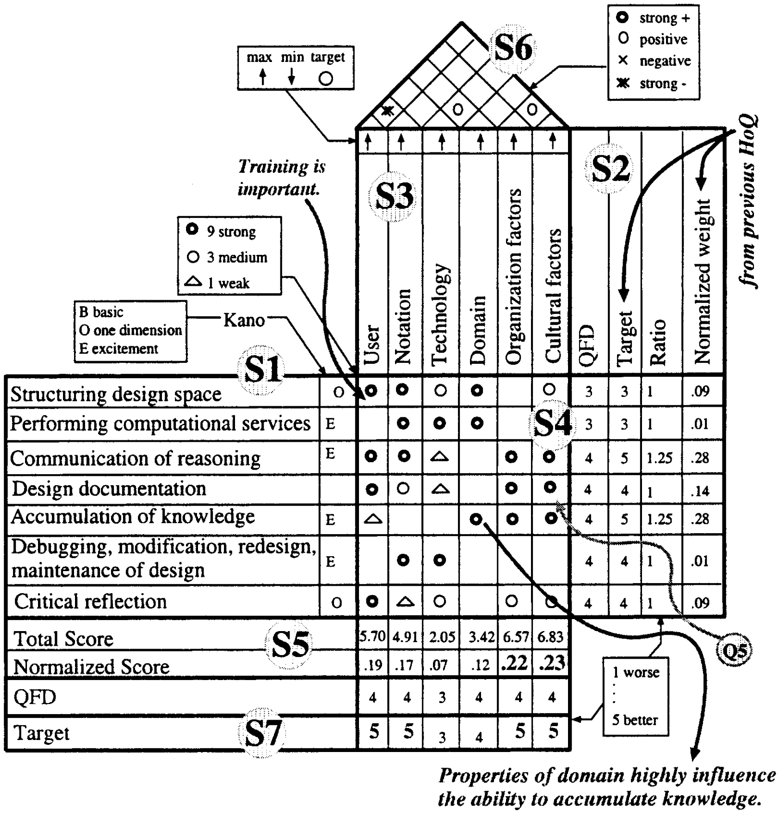


Fig. 11. HoQ: mapping DR tasks to project planning issues from Buckingham Shum [35].

communication with customers (Room 4) and gIBIS is rated average on *communication of reasoning* (3 in Room 5), how can it be rated poor on *communication with customers* (1 in Room 2)? This can be highlighted to the designer, who can reflect and argue that communication of reasoning is between two designers, which is different than communication between customers and designers. Hence, those three ratings are not contradictory and, in fact, may trigger refining the list of technical terms in room 3. This analysis can be done at any stage to check the compatibility between the content of rooms 2, 4 and 5.

S6: *Uncovering trade-offs.* This step identifies the trade-offs between the technical terms. These trade-offs can be relational statements between positions, which are more expressive than what tools such as gIBIS support. For example, *communication of reasoning* is positively correlated with *critical reflection.* The uncovering of trade-offs can be assisted by matrix analysis techniques (see Section 3.3).

S7: *Defining product goals.* The normalised scores of the technical terms and the trade-offs in room 6 determine the goals of the design in terms of target values of the technical terms. A technical term that receives high score should be improved. The amount of improvement depends on the difficulty of improvement and the trade-offs with other terms. We are developing a method to optimise the resource allocation for attaining the products goals [34].

In Fig. 10, the target goals provide the focus of subsequent design activities aimed at improving the ability of QFD to capture DR. From the HoQ, the focus should be on improving the capability of QFD to communicate reasoning, and to accumulate knowledge (as reflected by the normalized scores 0.28 of these capabilities). None of the other technical terms is targeted for improvement.

Figure 10 shows the translation from the general needs of DRC techniques to activities needed to support DRC. These activities are not users' needs, but researchers' perception of what facilities or activities must be supported to address the general needs. During the use of the HoQ, matrix data analyses can be applied, they are discussed in the next section. The use of the HoQ is often a cascading process, where goals from one house serve as the requirements of another [4]. Figure 11 shows a HoQ created to translate the target goals of the technical activities that a DRC tool should have from Fig. 10 into general project planning issues discussed by Buckingham Shum [35]. The process of using this HoQ is similar to the previous HoQ, except that the target values and normalised weights of the requirements are taken from the previous HoQ. The target goals in Fig. 11 reveal that *organisational factors* and *cultural factors* are the key contributors to DRC, while the *user* and *notation* factors are secondary. This is an interesting, albeit not a surprising, result.

3.3. Matrix Data Analysis

We use analysis tools available in KSSO, a repertory-grid knowledge acquisition tool [32], to analyse the data in Fig. 10. Repertory grid methods have been studied extensively [36]. These methods support the elicitation of important properties, whether desired or available, that DRC tools and methods should have. Therefore, they could serve as a means to establish the content of rooms S1 and S3 of the HoQ [37]. In addition, these methods can support an analysis of rooms 2 and 5 if the customer requirements and technical terms are considered as the attributes describing DRC techniques.

In this example, we only demonstrate the clustering (FOCUS) and the principal component analysis (PrinCom) that are available in KSS0. (We could have equally used similar methods in statistical packages.) The FOCUS results are shown in Fig. 12. Similar DRC techniques or attributes are grouped together. The grouping of the DRC techniques seems reasonable. The grouping of the attributes is less obvious. Insight from it can lead to uncovering correlations between technical terms and customer requirements. For example, *design documentation* correlates with *prevent loss of expertise*, and *critical reflection* correlates with *improved product quality*. These correlations can enhance our understanding of the domain, be captured in a relation diagram, or serve as input to refine room 4 entries. Upon inspection, the correlations from FOCUS appear in room 4 as strong or medium influences.

Another way to derive correlations is through the use of principal component analysis. Table 1 provides the correlations between the attributes calculated by PrinCom. The upper right part of the table reflects the correlations between the technical terms and the customer requirements, thus corresponds to room 4. The lower right part reflects the correlations within the technical terms, thus corresponds to room 6. Some correlations are rather different from those recorded in rooms 4 and 6 of the HoQ in Fig. 10, leading to examination and reflection. One exercise involves using the table values in the house for calculating the resulting normalised scores of the DR activities and the project planning issues. To do this, the correlations in the table were discretised according to Table 2. The discretised score is reflected by the number of overlines on entries in Table 1.

Note that, in the calculations, we are using discretised scores, although we could have used the raw correlation data. We are more interested in the qualitative influences than in precise magnitudes, which may not reveal much, and are the product of little data (i.e. 8 techniques described by 13 attributes). We can also conduct this exercise with different discretisation ranges, thus we are not concerned with their arbitrariness. When using the new correlations, the resulting scores of the technical terms are: (0.05, 0.07, 0.07, 0.13, 0.18, 0.21, 0.19), significantly different from previous results. In particular, the importance of *computational services, critical reflection* and *debugging* increased, and the importance of *communication of reasoning* decreased. The new scores of the project planning issues are: (0.20, 0.18, 0.14, 0.10, 0.19, 0.19), which are quite similar to previous scores, except for the importance of *technology*, which was increased in response to the increased importance of *computational services*.

Two interesting differences in the two sets of correlations are marked by **I9** and **I9′** in Fig. 10. We have pointed out that these correlations are in contradiction with the two other independent ratings in rooms 2 and 5, as shown in the figure. In contrast, the correlations according to Table 1 are weak for **I9** and none for **I9′**. Upon inspection, and as already mentioned before, the strong correlations are probably too strong: the fact that reasoning is communicated and knowledge is accumulated does not necessary lead to improving communication with customers or supporting company growth. These requirements also depend on other organisational and cultural factors.

Before proceeding in the use of other QFD tools, the correlations and other entries can be modified to reflect new insight about the domain. For the sake of simplicity, we continue with the existing normalized scores of the DR activities in the HoQ in Fig. 11 and Pugh concept selection tool.

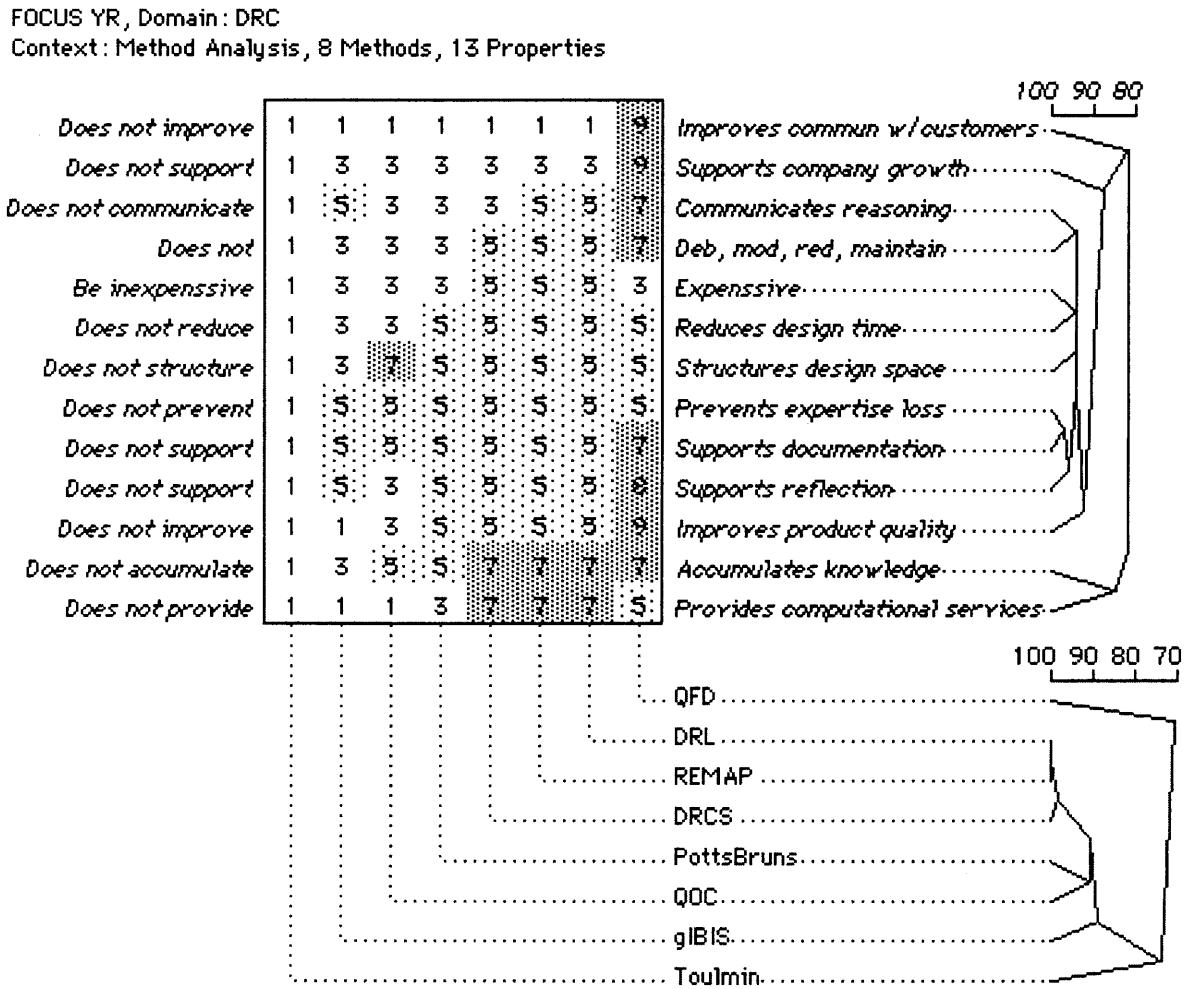


Fig. 12. Matrix Data Analysis: FOCUS clustering of HoQ data.

Table 1. Correlations between customer requirements and technical terms.

	Customer requirements						Technical terms						
	N1	N2	N3	N4	N5	N6	A1	A2	A3	A4	A5	A6	A7
N1	1.00	0.80	0.80	0.27	0.49	−0.80	0.67	0.81	0.61	0.79	0.92	0.82	0.80
N2		1.00	0.50	0.74	0.82	−0.43	0.51	0.73	0.65	0.74	0.82	0.89	0.82
N3			1.00	0.14	0.43	−0.71	0.76	0.43	0.65	0.91	0.76	0.65	0.73
N4				1.00	0.95	0.14	0.05	0.26	0.65	0.54	0.31	0.65	0.68
N5					1.00	−0.09	0.28	0.37	0.79	0.77	0.52	−0.79	0.84
N6						1.00	0.66	−0.78	−0.44	−0.54	−0.85	−0.65	−0.48
A1							1.00	0.40	0.39	0.67	0.78	0.54	0.41
A2								1.00	0.40	0.48	0.84	0.79	0.60
A3									1.00	0.83	0.62	0.83	0.89
A4										1.00	0.78	0.83	0.91
A5											1.00	0.89	0.72
A6		Symm.										1.00	0.89
A7													1.00

Table 2. Approximate ranges for discretising correlations.

Table correlation	Room 4 values	Table correlations	Room 6 values
1.00–0.85	9	1.00–0.80	Strong positive
0.85–0.70	3	0.80–0.60	Positive
0.70–0.55	1	Otherwise	None
Otherwise	0		

3.4. Pugh's Concept Selection

Upon completing the use of the HoQ and its analysis, we use Pugh's concept selection to develop a conceptual design of a new DRC technique. We use two sets of design criteria. The first set are the eleven issues raised throughout the paper. While not standard criteria, we can use Pugh's concept selection to asses whether and how well an alternative solution addresses a particular issue. The second set are the target goals from the HoQ in Fig. 11. First, we have to propose some solutions to the issues that subsequently will be synthesised into a design concept. The treatment of each issue can be done by a separate application of Pugh's concept selection; it can be captured by an IBIS-like diagram that records group discussions or by a QOC diagram that organises the results of such discussions.

The following list addresses each issue in turn:

I1: the incorporation of different process models into DR is a two step process. First, different process modelling methods have to be proposed and assessed, and secondly, the DR must be flexible enough to be tailored to each process model for its practical assessment. There are many available options for modelling processes, and their incorporation into the DR is a specific case of Issue **I2** with the addition of a computational facility to manage and enforce the model upon a particular design process.

I2: providing support for, and management of, tailoring DRC techniques in software is not trivial. Two options are available: providing mechanisms to define graph languages; or organising languages as objects in an object-oriented environment that supports inheritance and sub-classing. By using the second option, one can trace the evolution of languages, and better understand the role of experience and discipline in recording DR.

I3: the meaning of a term can be captured for future users only to a certain degree. The web of concepts linked to it and references to the source material provide additional context for understanding the meaning of terms. Such linkages can be implemented by hypermedia links.

I4: this issue is similar to Issue **I3**. To know whether two terms with the same label from different sources have similar meaning, the concepts need to refer back to their source material, thus giving users additional context for understanding their meaning. Hypermedia links can implement this.

I5: statements about, or references to, links could be represented if links become first class objects.

I6: complex relations between complex sub-diagrams to a conclusion can be represented if we allow large diagrams to serve as nodes in other diagrams.

I7: complex relational arguments could be represented if links become first class objects. In this way, a link may have as many incoming connections as needed to represent complex arguments.

I8: the identification of similarity between sub-diagrams detailing similar issues is a complex and often a social process. It can be initiated by posting an issue about the similarity between these sub-diagrams and arguing about it until the issue is resolved. Provision must be made to revise the resolution in response to new information or perspective.

I9: there is no way to prevent flaws in relation diagrams. In contrast, syntactic flaws can be prevented in tree diagrams by enforcing the DR language grammar. Detecting potential semantic flaws is much harder, however, there are some opportunities for detecting them. In relation to the HoQ we mentioned several such opportunities. First, when using the HoQ, for each requirement (Room 1) there must be at least one technical term (Room 3) that influences it through a value in Room 4. Otherwise, this requirement will not be satisfied. Secondly, we have already discussed the detection of mismatches between the three independent evaluations in the HoQ. When filling a HoQ, all such mismatches can be listed with different degrees of severity, and be stored until resolved by designers. Thirdly, different perspectives of the data can focus on issues that need fixing as discussed in Section 3.3.

I10: the manipulation of QFD tools on paper is cumbersome. Computerised versions of QFD tools with simple form filling capabilities can allow manipulating even 3D matrices. However, given that there are many types of matrices used for different purposes, and more may be invented, a means to create and manage them needs to be devised. A capability to define matrices by grammar would translate this issue into issue **I2**.

I11: in the HoQ, evaluations or relations (i.e. Rooms 2, 4, 5, 6) are limited to simple numeric rankings. In many cases in engineering we have more complex or mathematical relations available. We can easily point with a hypermedia link from a cell in these rooms to a source detailing such relations. Moreover, since we allow nesting of models as the resolution of issue **I6**, cells can point to arbitrary complex diagrams. We can also attempt to represent such relations explicitly [38], but the practical utility of all these ideas is unclear.

We select the baseline of designing a new DRC technique to be the concept of Computational QFD (CQFD), which is a partial software implementation of QFD with the ability to define graph structures and interlink them [39]. Based on this baseline, a second alternative that addresses the above issues is created: CQFDv2: CQFD with object-oriented graph-based modelling capabilities.

Pugh's concept selection is a means to select between these concepts. The method is very simple (see Fig. 13). The criteria are listed in the rows, and each column is a new candidate design concept. One of the concepts is selected as datum, and the remaining concepts are evaluated against it with respect to the criteria. The scores are simple: 'S' for same, '−' for worse than the datum, and '+' for better than the datum. A total score is calculated for each concept, allowing us to use different weights for each criterion. The weights can be set as the normalised scores of the criteria as calculated by the HoQ, or be estimated using other techniques. The concepts with the best scores are further developed in an attempt to remove the '−' scores while maintaining the remaining scores. After establishing several additional candidates, a new datum is selected and the process continues in a new phase.

CQFD: diagrams defined by graph languages
CQFD+AI support: AI used to provide computational facilities
CQFD v2: graph languages created as objects in some OOPL

		Concept						
		phase 1					*phase 2*	
Technical Criteria/Functions	weight	QFD (manual)	Computerized QFD	Hypermedia QFD (1st DATUM)	CQFD	CQFD+AI support	CQFD (2nd DATUM)	CQFD v2
1. process models and their tailoring	1	S	S		+	+		+
2. tailoring DR languages	1	S	S		+	+		+
3. meaning of terms	1	-	-		S	S		S
4. meaning of equal-label terms	1	-	-		S	S		S
5. statements about links	1	+	S		+	+		S
6. relations between diagrams and concepts	1	+	-		S	S		S
7. complex relational arguments	1	+	-		+	+		S
8. working with clusters of graphs	1	-	-		+	+		S
9. detecting flaws	1	-	S		+	+		+
10. computational support for matrices	1	-	S		S	S		+
11. increasing expressiveness of relations	1	-	-		+	+		+
General Criteria								
User	5	+	+		S	-		+
Notation	5	-	-		+	+		S
Technology	3	-	-		+	+		+
Domain	4	-	S		+	+		S
Organization factors	5	+	+		-	-		+
Cultural factors	5	S	S		+	S		S
Total score (technical)		-3	-6		7	7		5
Total score (general)		-2	+2		17	2		13

(General criteria weights: from previous HoQ target goals)

Fig. 13. Pugh's concept selection.

Two phases are depicted in Fig. 13. In the first phase a QFD tool with hypermedia linking facility is selected as the datum. The best concept CQFD is further developed into another variant taking into account some of the solutions to the afore-mentioned issues. A new datum is selected and another phase is performed. Considering technical merit, the final outcome is CQFD with object-oriented graph modelling. Subsequent to selecting a candidate design, the use of the PDPC and arrow diagrams can further improve it and capture additional DR. We did not use these tools in our design.

4. Reuse (Value) of DR Recorded by QFD Tools

Reusability is a key requirement of DR. Reusability manifests itself across projects, but also within projects. This can be demonstrated by listing the kinds of questions that can be answered about the information recorded in QFD tools, and the use of this information for reflecting-in-action.

In general, to answer such questions or record the process of answering them, we could treat each cell in rooms 2, 4, 5, 6 or 7 as the issue ‘determine this value’ and connect it to an IBIS-like model that articulates the argumentation, including nested models that point to the relevant relation or tree diagrams. This is easily realisable by the method proposed in the previous section.

Figure 10 includes question numbers shown as circles pointing to particular entries in the HoQ. The questions refer to the content of these entries. References to some of these questions appear in other figures as circles, denoting an area in the diagram that contain information relevant to its resolution. We consider the following questions:

Q1: why is the value of gIBIS evaluation with respect to structuring design space 2 as displayed in Fig. 10?

R1: the value 2 in the entry is connected via hyperlinks to the sources that establish it. One hyperlink can point to the part in Fig. 4 that suggests that since gIBIS is constructed during design, its payoff (in terms of understanding the space of possibilities) is limited. Another hyperlink can point to the part of Fig. 5 that adds that gIBIS can cause premature commitment, instead of encouraging the exploration of the complete design space.

Q2: why is QOC evaluation with respect to critical reflection 2?

R2: the hyperinks in this entry can point to the part in Fig. 4 that suggests that QOC is reconstructed after design, and that this increases payoff, but only in future design situations. The figure also suggests that recording DR separately from realisation prevents reflective exploration.

Q3: why is there no influence of computational services on product quality improvement?

R3: an entry with no value means there are no hyperlinks. Perhaps is was missed of judged to be so with no reference to other information. Nevertheless, if we follow the hyperlink under ‘Performing Computational Services’, we would get to parts of Figs 4 and 9 that explain the contribution of computational services towards improving product quality. This explanation may prompt us to suggest that we could introduce a strong correlation in the cell pointed to by Q3 in Room 4 of the house in Fig. 10. We could now explore what might have been the reasons for the present correlation, or we can modify it and check the sensitivity of the normalised scores in room 5 to this assignment; the new scores would be (0.08, 0.09, 0.26, 0.13, 0.26, 0.10, 0.08) – a large increase to the score of computational services, but nevertheless, one that still keeps them below the importance of *communication of reasoning* and *accumulation of knowledge*. If we are only interested in the most critical items, we do not need to spend further effort in resolving this question.

Critical reflection in design can be assisted by a quick ability to answer various ‘what if’ questions. A3 above includes such a question. Below are additional examples.

Q4: what would be the relative importance of DR tasks and project planning issues if we selected Potts and Bruns as a baseline technique for improvement?

R4: if Potts and Bruns becomes the baseline technique, the ratio column is changed to reflect the ratio between the target and the evaluation of Potts and Bruns model. This will change the normalised weight values and, subsequently, will lead to setting different target goals for the DR tasks and the project planning issues. Figure 14 details the normalised scores and target values for all the relevant parameters if Potts and Bruns or QOC were the baselines techniques. Clearly, given different baselines, different targets are set. It is not surprising that the targets for Potts and Bruns and QOC are almost identical. Interestingly, KSS0 clustering analysis in Fig. 12 found them to be very closely related. Perhaps the most striking difference between QFD and the other two techniques is the increased effort they require to *improve communication with customers*, which translates to increased effort to improve the *communication of reasoning* capability.

Q5: what would happen if in Fig. 11, the influence of *cultural factors* on *design documentation* would be 3 or 1 instead of 9?

R5: changing the value and recalculating the normalised scores will lead to the following scores of the project planning issues (0.21, 0.20, 0.13, 0.08, 0.18, 0.21), i.e. a slight reduction in the scores of *culture* and *organisation* factors, and a slight increase in the importance of the other factors. This outcome could be expected.

Q6: what would happen if the importance of *be inexpensive* would be 5 instead of 1?

R6: this modification leads to the following scores of the project planning issues (0.25, 0.24, 0.10,

	Baseline DRC Technique	QFD		Potts & Bruns		QOC	
		Normalized Score	Target	Normalized Score	Target	Normalized Score	Target
DR Requirements	Reduce design time	.20	4	.10	4	.13	4
	Improve product quality	.25	5	.21	5	.28	5
	Prevent loss of expertise (retirement, early leave)	.22	4	.11	4	.10	4
	Improves communication with customers	.17	5	.42	5	.37	5
	Support company growth	.13	5	.13	5	.11	5
	Be inexpenssive	.02	3	.01	3	.01	3
DR Tasks	Structuring design space	.06	3	.05	3	.05	4
	Performing computational services	.01	3	.01	2	.01	1
	Communication of reasoning	.29	5	.37	4	.34	4
	Design documentation	.15	4	.12	3	.12	3
	Accumulation of knowledge	.29	5	.24	4	.24	4
	Debugging, modification, redesign, maintenance of design	.11	4	.10	3	.12	3
	Critical reflection	.09	4	.09	3	.10	3
Project Planning Issues	User	.15	5	.17	3	.17	3
	Notation	.17	5	.18	4	.18	4
	Technology	.07	3	.06	3	.07	3
	Domain	.11	4	.09	4	.09	4
	Organization factors	.24	5	.24	4	.24	4
	Cultural factors	.25	5	.24	4	.24	4

Fig. 14. HoQ: Reuse of design rationale.

0.05, 0.16, 0.21), i.e. the human factors (culture, organisation and user) increase at the expense of the remaining factors. This is an interesting observation. If the cost of the DRC software package is important, it will have less computational services built into it, and the importance of users will become critical. In the long run, however, a reduction of labour might lead to a cheaper solution overall. This projection may prompt us to reduce the correlation between computational services and cost to −3 instead of −9, and to re-interpret the term *be inexpensive*.

5. Summary and Conclusions

Current argumentative DR languages are hardly or not at all used in design practice. A straightforward argument states that the use of practical design tools as a baseline for introducing DRC techniques into practice can improve this situation. If QFD tools, which are already DRC tools, are used as a baseline, we would have been able to implement even better DRC tools.

We illustrated the use of QFD tools in design by using them to improve the DRC capability of QFD by designing a computational QFD tool with object-oriented graph modelling capability. Such an environment allows hyperlinking between information object, and has the benefits that led to its selection at the end of Section 3.

In performing the improvement, the DR of this design was captured and several issues were raised. In designing the new technique, these issues were addressed. Thus, we illustrated how QFD tools serve as DRC tools, and how the DR they capture can be reused to answer various questions.

Some of the critical aspects uncovered by the use of QFD may be ‘surprising’, perhaps not their status but, nevertheless, their ‘quantitative’ priority. For example, although cultural or organisational issues are recognised as critical aspects underlying the adoption of technology in general, and of design tools specifically [29,40], it was not anticipated that they would turn out to dominate the success of a DRC technique as displayed quantitatively in Fig. 11 (i.e. determine 45% of the success of a tool). We wish to emphasise that in no way are these quantitative values given special or ‘objective’ status, rather they are used to focus attention on central design issues.

The process of using QFD tools in the design of a DRC technique forced us to better articulate the issues involved in building CQFD tools and solve them. Therefore, this design serves as an anecdotal evidence for the utility of using QFD tools as DRC tools. Another observation is that the use of QFD was time consuming, much like the use of DRC techniques. Nevertheless, whereas the use of DRC techniques presents designers with an overhead on top of their design work, and is thus often avoided, the use of QFD has passed the ‘acceptance test’, and is part of design work in many organisations. The utility of QFD has been documented in many case studies (e.g. see [8]). This utility has been boosting a cultural change in engineering practices, leading to internalising that spending the additional effort to train designers in using quality design techniques, and having designers use them result in better quality products in less overall time. Thus, we are not concerned that QFD will not be used, the question is how will QFD tools evolve to improve design practices.

The process of designing the improved DRC technique was captured partially by a manual simulation of the technique itself as displayed in the figures in this paper. These figures do not really demonstrate the strength of the approach. Some figures are overloaded due to space limitations, and others are missing. In addition, the interlinks between information objects is not visible. The value of the technique could only be appreciated through working with a computational implementation.

The proposed DRC technique addresses issues raised by Buckingham Shum [35] as an agenda for future DR research: it provides support for incremental flexible language construction, collaboration and complete integration in design, and not only with artifact representation. Furthermore, it will alleviate the difficulty of training by building upon techniques that are already being incorporated into industry practices. The implementation of the technique allows integration with other tools, and supports quick modifications or extensions as deemed appropriate for DRC.

The proposed technique addresses only some of the issues raised by Ullman [26]. For example, we discussed indirectly solutions to issues 6 and 11, and presented the position in relation to issue 13. The other issues must be dealt with in future refinements of the method following feedback from actual industrial use, rather than in purely theoretical studies. Otherwise, the practical utility of the proposed solutions could be compromised.

Finally, there is a close relation between some QFD and knowledge acquisition tools. They can be used synergistically, provided that this synergy proves useful in practice [37]. Work in knowledge acquisition should also be noted in relation to the last comment of this paper: the methodology aspect. It was Boose [41] who in 1989 analysed knowledge acquisition tools with a knowledge acquisition tool for the purpose of better understanding the domain. Similar practice should be continued and commended whenever possible. Researchers developing DRC techniques have always used text as the primary form of explaining their ideas. This paper departs from this practice by using a DRC technique to capture the rationale behind designing a DRC technique, thus creating a synergy between the theory of DRC and its practice. Such synergy can boost research tools towards becoming practically useful. Only by using DRC tools on our *own* designs can we, DR researchers, proclaim that these tools are useful to *other* designers.

Acknowledgements

I thank Brian Gaines and Mark Linster for the copy of KSS0 used in this study.

References

1. Buckingham Shum, S., Hammond, N. (1994) Argumentation-based design rationale: what use at what cost? Int. J. Human-Computer Studies, 40(4), 603–652
2. Conklin, J., Yakemovic, B. (1991) A process-oriented approach to design rationale. Human-Computer Interaction, 6(3–4), 357–391
3. Lee, J., Lai, K.-Y. (1991) What's in design rationale? Human-Computer Interaction, 6(3–4), 251–280
4. Akao, Y. (ed). (1990) Quality Function Deployment. Productivity Press, Cambridge, MA (original Japanese version from 1988)
5. King, B. (1989) Better Designs in Half the Time: Implementing QFD Quality Function Deployment in America. GOAL/QPC, Methuen, MA, 3rd ed
6. Clausing, D. (1993) Total Quality Development. ASME Press, New York
7. Cohen, L. (1995) Quality Function Deployment: How to Make QFD Work for You. Addison-Wesley, Reading, MA
8. QFD Institute (1999) Transactions of the eleventh symposium on quality function deployment. QFD Institute, Ann Arbor, MI
9. Toulmin, S. (1958) The Uses of Arguments. Cambridge University Press
10. Rittel, H. W. J., Webber, M. M. (1973) Dilemmas in a general theory of planning. Policy Sciences, 4, 155–169
11. Conklin, J., Begeman, M. L. (1998) gIBIS: A hypertext tool for exploratory policy discussion. ACM Trans. Office Infor. Syst., 6(4), 303–331
12. McCall, R. J. (1991) PHI: A conceptual foundation for design hypermedia. Design Studies, 12(1), 30–41
13. Fischer, G., Lemke, A. C., McCall, R., Morch, A. I. (1991) Making argumentation serve design. Human-Computer Interaction, 6(3–4), 393–419
14. Potts, C., Burns, G. (1988) Recording the reasons for design decisions. Proc. 10th International Conference on Software Engineering, pp 418–427
15. Garcia, A. C. B., Howard, H. C., Stefik, M. J. (1994) Improving design and documentation by using partially automated synthesis. AIEDAM, 8(4), 335–354
16. Ramesh, B., Dhar, V. (1992) Supporting systems development using knowledge captured during requirements engineering. IEEE Trans Softw. Eng., 18(6), 498–510
17. Reeves, B., Shipman, F. (1992) Supporting communication between designers with artifact-centered evolving information spaces. In: Turner, J., Kraut, R. (eds.), CSCW '92 Proceedings of the Conference on Computer-Supported Cooperative Work, Toronto, Canada. ACM Press, pp 394–401
18. Ruecker, L., Seering, W. P. (1991) A dialectical reasoning system for design documentation. Design Theory and Methodology – DTM'91, Miami, FL. ASME, pp 271–278
19. Gaines, B. R., Shaw, M. L. G. (1995) Concept maps as hypermedia components. Int. J. Human-Computer Studies, 43(3), 323–361
20. Levy, S., Subrahmanian, E., Konda, S. L., Coyne, R. F., Westerberg, A. W., Reich, Y. (1993) An overview of the *n*-dim environment. Technical Report EDRC-

05–65–93, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA
21. Reich, Y., Subrahmanian, E., Cunningham, D., Dutoit, A., Konda, S., Patrick, R., Westerberg, A., and the *n*-dim group (1999). Building agility for developing agile design information systems. Res. in Eng. Design, 11(2), 67–83
22. MacLean, A., Young, R., Bellotti, V., Moran, T. (1991) Questions, options, and criteria: elements of design space analysis. Human-Computer Interaction, 6(3&4), 201–250
23. McKerlie, D., MacLean, A. (1994) Reasoning with design rationale: practical experience with design space analysis. Design Studies, 15(2), 214–226
24. Lee, J. (1990) SIBYL: A tool for managing group decision rational. Proc. CSCW 90, ACM Press, pp 79–92
25. Klein, M. (1993) Capturing design rationale in concurrent engineering teams. IEEE Computer, 26(1), 39–47
26. Ullman, D. G. (1994) Issues critical to the development of design history, design rationale and design intent systems. DE-94 68, Design Theory and Methodology – DTM '94. ASME, pp 249–258
27. Heliades, G. P., Edmonds, E. A. (2000) Notation and nature of task in comprehending design rationale. Knowledge-Based Systems, 13, 215–224
28. Oinas-Kukkonen, H. (1998) Evaluating the usefulness of design rationale in CASE. Euro. J. Infor. Syst. 7, 185–191
29. Reich, Y. (1994) What is wrong with CAE and can it be fixed. Preprints of Bridging the Generations: An International Workshop on the Future Directions of Computer-Aided Engineering, Department of Civil Engineering, Carnegie Mellon University
30. Mizuno, S. (ed). (1988) Management for Quality Improvement: The Seven New QC Tools. Produtivity Press, Cambridge, MA (original Japanese edition by JUSE from 1979)
31. Pugh, S. (1981) Concept selection – a method that works. Proc. International Conference on Engineering Design (ICED-81), Heurista, Zurich, pp 497–506
32. Gaines, B. R., Shaw, M. L. G. (1993) Eliciting knowledge and transferring it effectively to a knowledge-based system. IEEE Trans Knowledge and Data Eng., 5(1), 4–14
33. Suh, N. P. (1990) The Principles of Design. Oxford University Press
34. Levy, E. (2000) Software quality function deployment. Master's thesis, Department of Solid Mechanics, Materials and Structures, Faculty of Engineering, Tel-Aviv University
35. Buckingham Shum, S. (1996) Design argumentation as design rationale. In: Kent, A., Williams, J. G. (eds.), The Encyclopedia of Computer Science and Technology. Marcel Dekker
36. Gaines, B. R., Shaw, M. L. G. (1993) Knowledge acquisition tools based on personal construct psychology. The Knowledge Eng. Rev., 8(1); 49–85
37. Reich, Y. (1996) AI-supported quality function deployment. In: Ein-Dor, P. (ed.), Artificial Intelligence in Economics and Management. Kluwer, pp 91–106
38. Ramaswamy, R., Ulrich, K. (1993) Augmenting the house of quality with engineering models. Res. in Eng. Design, 5, 70–79
39. Boose, J. H. (1989) A survey of knowledge acquisition techniques and tools. Knowledge Acquisition, 1(1); 3–37
40. Reich, Y. (1996) Computational quality function deployment is knowledge intensive engineering. In: Tomiyama, T., Mantyla, M., Finger, S. (eds.), Knowledge Intensive CAD. Chapman & Hall
41. Reich, Y. (1992) Transcending the theory-practice problem of technology. Technical Report EDRC 12–51–92, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA