# Constraining and Creating Solutions - reflections on the analysis of early design

Chris Roast

Culture, Communication and Computing Research Institute
Sheffield Hallam University
Sheffield, S1 1WB
United Kingdom
c.r.roast@shu.ac.uk

**Abstract.** This research explores how an existing analytic framework (the cognitive dimensions framework) for interactive digital design reflects knowledge relevant to exploring the design space. The work examines this idea through the analysis of the transcripts of three digital design collaborative workshops run as part of "Studying Professional Software Design"[1]. Expert deliberation within these workshops is assessed and related to the analytic framework. The cognitive dimension framework has not been applied to observational data of this sort before. However, the approach described in this paper appears to provide a viable means of analysis. In conclusion we demonstrate that approaching observational data in this manner is not highly complex but is sufficient to provide useful insights. Reflections from the resulting analysis shed light on the interests and tensions evident in early stages of digital product design.

**Keywords:** user centred development, early design decisions, cognitive dimensions.

## 1    Introduction

This paper describes an exploration of the design process for digital systems, motivated by the analysis of data taken from three professional digital design workshops illustrating early concept formation [1] and collaborative research with professional developers on a live commercial project [2]. We are particularly interested in the design space of interactive digital design and how well that space is navigated and understood during the process of developing a conceptual design. The motivation for this work is based on an interest in understanding expertise in the domain of interactive digital design, exploring the implications for how design might be improved or supported.

Here we consider design expertise to reveals itself in terms of being able to short-circuit what would otherwise be time consuming problem-solving expeditions. It is

---

[1] For more information on these resources the reader is direct to: http://www.ics.uci.edu/design-workshop/videos.html.

very common for large scale design problems to be addressed through tackling components with them. Hence managing complexity is achieved through structures that modularize or scope specific challenges [3]. However the digital design space embodies a massive range of complexities. Jackson [4] provides a rich account of how abstractions and perspectives are core to managing complexity in contemporary software projects. Despite that, working with and effectively using such structures is non-trivial. The mechanisms to cope with the complexity are not comprehensive and they thus fail to deal adequately with some design requirements. One significant contributory factor is the persistent opacity of digital artefacts. The structures employed to manage complexity are intrinsically abstract and poorly understood by non-technical stakeholders. In addition, while abstractions and structures can enable constructive development they can always be usurped both technically and by the emergent user needs. System requirements of interactive devices are frequently realized via their behaviours and these are hard to both specify and comprehend prior to encountering them via experience. A sample of research in the area of the psychology of digital programming shows that the ambition of wishing to engineer digital systems can be undermined by the fact that even simple digital programs can be difficult to comprehend [5]. In particular specific highly promoted techniques aimed at addressing some elements of perceived complexity have been shown to undermine their overall rationale [6].

The view taken here is that the need to structure and manage the complexity of the digital design space can be interpreted as a means of representing the space in a manner that is amenable for navigation and exploration. As such the manner in which the structure is imposed is attempting to serve to two purposes: (i) to make it easier for human designers to work with, and (ii) make it appropriate for technically meeting the requirements of a particular design problem. We are interested in the first of these - the fact that human factors plays a role in determining the way in which the design space is explored. Hence, when faced with a range of technical approaches in a design process, the choice of approach will not only be governed by the technical quality but also by the ease with it can be understood and applied. The quality of the representations employed and the experience and expertise of the designers will inform such a choice. It is interesting to see whether: expertise applied in a design setting demonstrates insights into the digital design space, thus reducing the likelihood of poor choices; how a design representation supports that process; and, the extent to which choices address the technical requirements or client needs.

## 1.1 An analytic framework

The chosen analytic framework for considering human factors and the exploration of digital design space is that of "cognitive dimensions" [7]. The cognitive dimensions framework provides a set of concepts that have been used as a means of capturing generic aspects of human interaction with complex information structures, with aim of emphasizing recurring factors pertinent to design quality [8]. This framework is unique in that: (i) it has been developed from the examination of the hard problems that arise with information systems such as the use of representations to define and comprehend complex behaviours, and (ii) it has been articulated in terms that do not

presuppose human factors expertise. In doing so, the framework appears to be appeal as a means of easily capturing important observations about concepts are expressed and developed.

To illustrate this point we briefly introduce one of elements of framework - "**hidden dependencies**". As its name suggests, this refers to the fact that digital artefacts can manage masses of inter-related data items which means changing one item can influence many others. Despite users and developers being fully aware of such dependencies, they can still manifest themselves un-expectedly. Tools and notations used when designing and/or using a system can be judged in terms of how they form, expose and manage such dependencies. Design often uses diagrams and structures to capture and circumscribe both data and functional dependencies. While these make explicit some dependencies others can be hidden and still need to managed by, say, cross-checking design changes in each of a number of views. Our proposal in this paper is that digital design experts are likely to have experienced re-current issues arising from working with concepts such as hidden dependencies and will therefore be prepared for dealing them. Designers with greater expertise may be more attentive to managing dependencies.

To date the concepts from cognitive dimensions framework have been used constructively to frame or drive reflections, hence "hidden dependencies" provides a way of seeing a particular problem [2]. By contrast, the work reported faces two challenges. First, whether the interesting features captured by the dimension framework relate to professional design expertise and experience, and second, whether such a relationship is evidence in "raw" observational data. To address the first of these, for this study, we are pre-supposing that expert experience based design behaviour will, in part, match the insights that the framework is designed to encourage. While this premise is speculative, it is not unrealistic especially since the framework is based on extensive research experience examining human behaviour in relationship to technology supported development. However, we'd not claim that such a match is a basis from assuming the framework reflects expert mental processes. The second challenge is to employ the dimension concepts in the analysis of observational data, and our approach is described below, see sections 3 and 4.

### 1.2 The workshop data

The data used for this work is from expert developers engaged in common a design task. This was developed as part of project aimed at generating an open research resource based upon collecting a foundational set of observations and insights into software design[2]. The task examined was to develop a preliminary user interface design and also data model and architecture for a given "brief". The brief was for a traffic simulation system explicitly for use in an education context. The simulation would used by tutors and students to explore the influence of different traffic signal protocols upon traffic flow. Rich contextual detail was provided regarding what can be assumed of the simulated traffic system and what functionality could be assumed

---

[2] Comprehensive details of this project, the developer tasks and the data are available at: http://www.ics.uci.edu/design-workshop, also see [1].

to already exist. In addition, the set task required both the completion of technical and design details in limited time.

Three separate pairs of professional developers worked on the given task for between one to two hours with limited additional support. They were required to manage their own activities while completing the task and thus had to balance between technical concerns and those of design. Their discussion and use of a whiteboard was recorded and transcribed - this data is freely available from the project coordinators and other research exploring the same data sets is the subject of a special issue on Design Studies [1]. In summary: the workshops lasted between one hour and two hours and the transcripts of discussions being between 6,000 and 12,000 words long.

## 2   The Cognitive Dimension Framework

The Cognitive Dimensions Framework is a relatively informal means of examining the space of design alternatives from the perspective of exploring the factors that can influence utilization of an artifact - one such factor "hidden dependencies" was introduced earlier. The framework arose primarily from the fields of human factors and human-computer interaction, and the examination of users working with digital languages, notations and representations. This is of relevance to examining design and development since it has been motivated by the examination of languages and representations by which digital systems are themselves are defined. A common feature of the representation examined has been their use for defining activities in the future, such as: (i) expressing code that subsequently operates autonomously; or (ii) developing, say a, set of rules that subsequently auto-process unwanted emails. Unlike many more traditional user interface assessment and analysis approaches, the framework focuses upon important factors related to effective use, without presupposing their relative importance. This enables its use as a basis for discussion and reflection in a wide range of contexts. Hence, while the framework was initial motivated by examples taken from structured visualisations, it has developed to be applied to a broader range of information artifacts. Topics examined using the framework, include: textual and visual programming languages; APIs [9]; specification notations [10]; spreadsheets [11]; musical notations [12]. Green et al. [8] provides a summary of the range of work employing the framework and its development. Recently we have employed a simple tabular format for reporting based on the framework that has been successfully used to facilitate co-operative assessment of complex systems [2].

The dimensions of the framework are discussed and illustrated thoroughly in [13], below we outline some of the dimension concepts and suggest their potential role in design. It will be clear from these brief introductions the concepts are closely associated and to some extent overlap.

**Premature commitment** – this concerns the idea that systems and notations can demand that users make decisions before they are ready to. Although this behavior may be "demanded" it might not in fact be technically necessary. By contrast while

such behavior might be seen as unsuitable, it might be purposefully designed to promote a greater diligence or responsibility on the part of the user. An example of premature commitment might be in a formatting rule for inserting an image in a document, using the rule might demand that the image file is specified and exists, when in fact a placeholder could be provided.

**Viscosity** – this concerns the relative complexity of changing information represented in a system. This relates closely to common notions of efficiency in usability. However a key reflective point promoted by the concept is that while a notation tends to make some changes easy, others may be purposefully made complex or hard. In the example of formatting rules, a relative positioning rule (align A with B) can be compared with an absolute positioning rule (find the position of B and set A to that position). Assessing their viscosity would encourage the consideration of what changes to rule uses are most likely to be supported by these two alternatives and whether they are both needed.

**Visualisation and juxtaposition** – this concerns information outputs that the user may use in combination, and the difficulty or ease of combining them. In the simplest chase information needs are met by presenting information juxtaposed making human perceptual inferences easy, such as comparing values on a histogram. More often than not all possible information needs can be satisfied and the user might then have to navigate between sources, possible even switching between applications. Using the example of formatting rules, we can envisage the benefit of being able to see the rules that influence the same element juxtaposed with one another.

**Abstraction** – this concerns the potential to employ abstractions within a systems and the ease with which they can be used. Abstraction normally supports working with general classes or groups of entities and not specific individuals. It is used productively in many settings within software engineering. By contrast some systems impose pre-defined abstractions that users then have to assimilate. As with viscosity, the concept of abstraction is used as a means of reflecting on the nature of support that is provided. An example of abstraction with formatting rules would be how groups of rules serving a collective purpose might be managed as a single composite entity.

**Hard mental operations** – this concept has close associations with Abstraction, it is concerned with whether the operations available to users of a representation are easy or hard to understand. For this it is necessary to envisage the manner in which a user will think about achieving a particular objective and whether the operations provided by the system match them. In the case of a set of formatting rules a hard mental operations would reveal themselves when updating a single rule rarely serves a meaningful purpose. One can envisage a number of updates to different rules being needed in order to achieve a required effect.

**Provisionality** – this concept focuses upon how speculative or incomplete work is supported. In system supported activity it is frequently the case that the user is constrained to completing specific operations or procedures, while this can be helpful, it can impair speculative activity or experimentation. Provisionality thus promotes the consideration of what might benefit from being transient or incomplete in order to encourage effective use. In the case of formatting rules, high

provisionality could mean ill-formed, incomplete or inconsistent rules do not "break" the system but are worked with.

## 3    Applying the framework

To employ the concepts from the cognitive dimensions framework in examining observational data, it is necessary to identify how features of the observational data related to the concepts of interest. To achieve this we rely upon the premise that points in the workshop discussions where decisions are made are likely to provide explicit indications of participants' expertise. Such "decision points" were urged to be locations where expertise and insight are most likely to reveal themselves and thus be appropriate candidates for finding relating to dimension concepts. Hence, specifics of deliberations at these points would be mapped to concepts provided a focus for the more detailed assessment of the data.

Following a preliminary assessment of the workshop resources a prior assessment of the dimensions was conducted mapping them to rationale topics that likely to be directly evidenced in design deliberations.  In summary, table 1 indicates the relative relevance of factors for each of dimensions considered. This provides a means of relating what is being discussed during design deliberations to the dimensions that are potentially applicable.

**Table 1.** Relating deliberation topics to the framework concepts (\*\*\* = highly relevant; \*\*=relevant; \*=possibly relevant)

|  |  | Viscosity | Visibility | Hidden Dependencies | Hard mental operations | Provisionality | Premature Commitment | Abstraction |
|---|---|---|---|---|---|---|---|---|
| Mental effort |  | * |  | ** | *** | *** | *** | ** |
| Physical effort |  | *** | *** | * |  | *** | ** | ** |
| Actively – doing things | Creating |  | * | *** | ** | ** | ** | * |
|  | Modifying | *** | * | *** | ** |  |  | * |
| Non active (ie. navigating) |  | ** | *** | * |  |  |  | * |
| Meaningfulness |  |  |  | ** | *** | * | ** | *** |

The different strengths of mapping shown in table 5 were determined from the critical assessment of cognitive dimensions framework and its prior use.  For instance, the dimension of 'Viscosity' focuses upon the complexity of making changes and thus we'd expect this to be evidenced in discussions of the physical effort required to when making changes for specific purposes. Thus discussions focused upon the modification of application data are likely to also be highly relevant. Less relevant will be user navigation since changing data in realistic setting can incur navigation. Similarly, when assessing 'Abstraction' it can be argued that it is predominantly

concerned with support for managing aggregates and groupings. Evidence of this concept in design discussions would include: discussion of the meaningfulness of categories; discussion of the physical and mental effort associated with operations upon and navigating abstractions especially user created abstractions; less closely related is the influence of abstraction when acting on or navigating application data. This type of assessment was conducted for each of the dimensions, linking them to broad topic categories that are more likely to be evidenced in design discussions.

The rationale for creating and using the mapping in table 2 is predominantly to provide a method for managing unstructured observational data. This benefit is qualified by the fact that the mapping could itself weaken the association between observed data and its interpretation. Although this mapping may serve as a guide as to what dimensions are "evidenced", the key indicator of expertise relates to the extent to which a discussion embodies some reflection upon alternatives and an awareness of the longer term consequences of particular decisions.


## 4   Analysis and outcomes

Following the initial review of the resources, a strategy to identify significant points in the workshop was developed. The primary rationale for this was to provide a focus for more detailed assessment. The approach taken was to read through the workshop transcriptions and highlight segments of the discussion at which it appeared that constructive decisions were made – i.e. decisions add detail or substance to the design based upon rational assessment. Key points in the transcripts where explicit alternatives were being articulated and assessed were taken to be of highest potential relevance; followed by, points involving the implicit assessment of alternatives (these might even be where workshop participants make unilateral assertions about a design option).

Selected foci were each assessed in terms of the reflective depth of the consideration of the topics. This involved reading the relevant section of transcript and also viewing the video record of the same section. At this stage of the analysis it became clear that the motivation for the decisions fell into two broad categories based upon the primary intended beneficiary of the decision: 'external users' - the intended users of the final design (tutors and students using the simulation tool for educational purposes); 'internal users' – those involved in subsequent technical development (software developers and designers). These two categories were used to factor the initial assessment of decision points.

**Table 2.** Assessment of workshop 1 focused upon discussions at selected key points:

| Locus of decision timestamp | What participants discussed and basis for their argument | User focus |
|---|---|---|
| [12:01.1] | Participants agree on there being no need to model road traffic lanes. Argument based on avoiding complexity, dependencies and whether lanes are required by the brief.<br>Rationale: avoiding **physical effort** and implicit **mental effort**, making code **creation** easier.<br>Dominant framework concepts: **Provisionality** and **Premature Commitment** | Internal |
| | The implications for tutors and student users discussed.<br>Rationale: **mental effort** and **physical effort**, especially when **creating** a configuration.<br>Dominant framework concepts: **Provisionality** and **Premature Commitment** | External |
| [23:02.8] | Revisiting the modelling of the lanes with a focus upon the integrity of the simulation (e.g. traffic queuing).<br>Rationale: reducing **mental effort** and **physical effort** and **meaningfulness**<br>Dominant framework concepts: **Abstraction** and **Premature Commitment** | Internal |
| | Rationale: **meaningfulness** for the tutor<br>Dominant framework concepts: **Abstraction** | External |
| [49:33.4] | Revisiting more details about the operation of the lanes model.<br>Rationale: the "perfect" simplification reducing **mental effort**.<br>Dominant framework concepts: **Premature commitment** | Internals |

The outcome of the assessment of the decision points and their mapping to cognitive dimension concepts is summarized in tables 2, 3 and 4. Linking this assessment back to the cognitive dimensions framework the rationale supporting decisions has been assessed in terms the relevant topics used in table 1. Working backwards from that mapping, it is possible to associate specific decision points with the dimensions of most relevance. This has been an analytic process based upon the assessment of the workshop discussions at the point of interest and the relevance to the dimensions. Where possible a single dominant dimension has been identified, though on occasions the mapping and the workshop transcript could not be used to effectively to identify a single dimension. However as we have pointed out the potential for overlaps between the different concepts is high, and when necessary critical judgment was applied to identify a dominant dimension in specific cases. The outcome is indicated in Table 5.

**Table 3.** Assessment of workshop 2 focused upon discussions at selected key points

| Locus of decision timestamp | What participants discussed and basis for their argument | User focus |
|---|---|---|
| [0:47:49] | Debating how best to implement the propagation of traffic flow information.<br>Rationale: referring to patterns of processing, **meaningful** software engineering concepts.<br>Dominant framework concepts: **Abstraction** | Internals |
|  | Rationale: an architecture offering users an easy link between **changing** configurable elements and **seeing** the consequences.<br>Dominant framework concepts: **Viscosity** | External |
| [1:01:10] | Discussion of where best to position behavioural rules in the data model.<br>Rationale: size and complexity of alternative options imposing **mental effort** making **modifications** hard to manage. Software pattern approaches considered as **meaningful** concepts.<br>Dominant framework concepts: **Hidden dependencies** | Internal |
| [1:17:36] | Discussing a conflict between cars initiated with a behavior when in that behavior might get **modified**. [Unresolved due to time limitations] | Internal |
| [1:26:45] | User interface design discussed, considering providing: pause, fast forward and re-set style controls, and showing actual cars and their movement.<br>Rationale: prior decisions regarding technical model dominate considerations.<br>[Unresolved] | External |
| [1:31:56] | Discussing various other interaction design options.<br>Rationale: **mental** and **physical effort** when a viewing simulation (**passively** exploring simulation as it runs).<br>Dominant framework concepts: **Visibility** | External |
| [1:41:44] | Discussing how intersections can inferred from overall road grid (and the implications) of that.<br>Rationale: **mental** and **physical effort** when **creating** the tool to cope with the problem.<br>Dominant framework concepts: **Hidden dependencies** | Internal |

**Table 4.** Assessment of workshop 3 focused upon discussions at selected key points

| Locus of decision timestamp | What participants discussed and basis for their argument | User focus |
|---|---|---|
| [0:21:09] | Discussion how junctions are created and configured by users. Rationale: reducing **physical** and **mental effort** when **creating** a road layout. Dominant framework concepts: **Premature commitment** and **Provisionality** | External |
| | Discussion how junctions might be manipulated by users. Rationale: reducing **physical effort** and **mental effort** when **manipulating** and when **viewing** a road layout. Dominant framework concepts: **Hidden dependencies** and **Visibility** and **Viscosity** | External |
| [0:41:13] | Discussing junction behaviour in detail with an emphasis upon the representation and configuration of traffic light behaviour Rationale: **Modifying** and **passively** viewing configurations considered, as well as **meaningfulness** for users and the **physical effort** involved. Dominant framework concepts: **Hidden dependencies** | External |
| [1:14:03] | Discussing how to show and edit junction details (popping-up versus in side bar). Rationale (as above) some choices deferred to "prototype". | External |
| [1:27:48] | Discussing differentiating modes of use: configuring; experimenting and running. Rationale: High level concepts of purpose (**meaningfulness**). Dominant framework concepts: **Abstraction** | External |
| [1:31:05] | Discussion of feedback and visualising feedback and providing tabular results. Rationale: reducing **mental effort** and allowing results to be **navigated** with little **physical effort** Dominant framework concepts: **Visibility** | External |
| [1:33:49] | Considering the potential of real-time configurable simulation and value of seeing immediate cause and effect. Rationale: allowing **modifications** and **passively** interpreting the results to infer **meaningful** relationships Dominant framework concepts: **Visibility** and **Hidden dependencies** | External |
| [1:47:53] | Summarise points requiring clarification | Internal |

## 5 Findings

An overall reflection on this the analysis shows that each workshop tended to have a predominant focus upon either "internal users" or "external users", and that the focus tends to relate to different cognitive dimensions. Design discussions focused upon the needs and requirements of end users appeared to be assessed in terms concerned with

the immediate complexity of seeing information and working with it. This lead to a focus upon issues concerning: Visibility, Viscosity and Hidden dependencies. By contrast considerations of the internal issues, such as the suitability of a design decision for subsequent technical development, appeared to emphasize: Provisionality; Premature Commitment and Abstraction. This grouping of the dimension concepts is similar to that found in a earlier study conducted in which professional developers were engaged in co-operative assessment of one of their software application's with motivating questions derived from the same framework [2].

## 5.2 General observations

The briefing for the workshop asked for the development of both technical details (such as a data model) and a user interface design within a limited time frame. This created a tension in the workshop of meeting both aims. Despite this, the interplay between the technical and user perspective was only rarely considered. In particular there was one occasion of an unforeseen conflict between technical design and probable user needs. The technical design provided an abstract model of traffic flow, while user needs were judged to demand a fine grain simulation.

**Table 5.** Outline mapping of cognitive dimensions to workshop data indicating the focus upon internal or external user needs.

|  | Workshop | | |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| Viscosity |  | e | e |
| Visibility |  | e | e e e |
| Hidden Dependencies |  | i i | e e e |
| Mental operations |  |  |  |
| Provisionality | i i |  |  |
| Premature commitment | i i i |  | e |
| Abstraction | i i | i | e |

With the discussions in two of the workshops focusing mainly upon the internal requirements, it is of interest that the participants frequently referred to themselves as engaging in future work on the project. They were positioning themselves as "developers". It appears that participants in these cases had a mutual understanding of the implications of a number of decisions — in-depth discussions were not common and summary justifications appeared to be adequate. For instance, on occasions there was consensus that a particular decision would "simplify" a solution, without a need for further explanation. Hence, it seems that workshops with a good common knowledge of technical possibilities pursued that route, and in doing so did not need to openly deliberate about specific technical choices. This can approach also go hand-in-hand with a reticence to explore factors that are unfamiliar or not well understood. Hence, the openness of the brief could be seen as providing too much freedom. In particular the lack of a detailed user perspective meant the workshop was more reliant upon speculatively envisaging users. A dominant illustration of this for the workshop

was the response to the lack of detail about the educational context set-up in the brief. The specific teaching and learning style to be used is not articulated, and hence the brief does not promote design decisions impinging on teaching and learning. Expertise in that area may breed the innovative exploration of learning styles, however pedagogic factors were only addressed partially and indirectly. This is an illustration how design options are not thoroughly explored.

Another distinction between workshops and their user focus are the apparent development styles. Those who focused upon internal user requirement worked consistently with the attitude that: (i) the right technical decisions are needed to ease future work; (ii) subsequent technical develop would follow good practice. By contrast the opportunity for adopting an "agile" or iterative development approach to refining a solution was only alluded to in the externally focused workshop (workshop 3).

## 5.3 Framework based assessment

The relationship between the cognitive dimensions identified in our analysis and the design deliberations can be examined in more detail. In particular we can assess the extent to which factors implied by the dimensions are evident in the discussions in the workshops.

**Internal assessment** Premature commitment, provisionality, hidden dependencies and abstraction are significant in the internal requirements discussions.

The workshop requirement of developing a data model was clearly considered as a point of high commitment, and significant effort went into getting it right. There were a few occasions when the technical details were in effect deferred by embodying a general structure in the model - an example of this is the adoption of a software engineering patterns. In terms of the cognitive dimensions framework this suggests that a data model in itself is too committing, promoting technical deliberation prior to full understanding overall needs.

Hidden dependencies appear in the discussion of technical decisions when the discussion uses mini-scenarios to illustrate complexity. The features of hidden dependencies and ways of managing them were not explored directly by the participants. Were this style of justification by illustration not adopted, the discussion could easily be classed as being focused upon premature commitment as opposed to hidden dependencies.

Abstraction stands out as being a moderating force when considering "simple" models - a key factor being the "meaningfulness" of a proposed alternative. In the data analysed two domains of meaningfulness are employed: meaningful in terms of external requirements (such as the realisation of "traffic queues") and meaningfulness in terms of familiar software engineering approaches (such as data models and software engineering patterns).

In conclusion the demands of developing a data model appear to be highly committing and potentially disruptive to the balancing of internal and external considerations. On the basis that a data model is needed it appears that concepts such as software engineering patterns provide a means of delaying commitment to

technical details. While this could have provided a solid technical approach from which external factors could be explored but this opportunity was not taken.

**External assessment** Visibility, Viscosity and Hidden dependencies are the dominant dimensions mapped to when considering external users needs. The inter-relationship between viscosity and visibility is close with the main distinguishing factor being whether the requirement concerns users passively examining information or actively engaged in changing data or configurations. In the majority of cases the discussion focused upon possible user interface options, to ensure particular tasks are easily performed and required data is visualised usefully. What these two dimensions would bring to such discussions is a focus upon the side effects of particular solutions in order to encourage critical reflection. So in the case of visualising data, screen area is a limiting factor which naturally promotes the consideration of exactly what, and how much, information can be realistically be simultaneously displayed in a coherent manner. On a few occasions the consideration of such side effects are evidence. This is most obvious in workshop 3 where the whiteboard was used to draw a composite of display design details and application data. In this case the board provided a limiting factor of sorts.

The mapping to hidden dependencies arises in one case from the domain topic of the workshop. The simulation system being envisaged serves an educational purpose, and the workshop participants recognize that learning in that setting often arises from finding inter-relationships.

The other case of hidden dependencies is related to other dimensions and arises when the configuration of a simulation is being considered. Here the motivation was to ease the management of the data structure needed to define a simulation. The visual approach to addressing this problem, by using the whiteboard extensively, encourages the consideration of solutions where the complexity is visually evident and thus potentially less hidden.

### 5.4 Recommendations

Having examined early stage software design workshops and identified potential links with the cognitive dimensions framework. A number of conclusions can be drawn about the exploration of the design space. Some arise from reflections on the workshops themselves while others are the consequence of examining how the cognitive dimensions framework could inform this type of design activity.

**Get into problems** Early stages of development suffer from often being too open. This lack of constraints means that design details can be proposed and agreed with little consideration. Only when enough detail is established do decisions demand a more thorough and careful examination and justification.

Approaches to avoiding decisions in a vacuum encourage inherent problems to be confronted early. Two examples of this in the workshops studied are: (i) to promote the consideration of the problems arising from different perspectives (such

as internal and external); (ii) exaggerate factors to constrain the problem space, such as requiring a specifically small screen size.

**Feed creativity** Initial design concepts lack details. For a design team to work effectively with them they must be confident to work creatively in order explore a design space. This requires the setting to be one in which the ethos is to find interesting ideas and avoid "solutions".

Design representations should enable provisonality or variablity so as to limit their treatment as definitive representations. Similarly representations should be understood as being part of a process of iterative exploration, and hence they are best viewed as transient representations.

Too much common knowledge within a team can discourage the exploration of alternatives. This is exacerbated if they are exploring aspects of system use and operation outside their collective expertise.

**Less selectivity** One of the factors that appears to limit effective exploration of the design space is a tendency to focus on some facets of a problem and ignore others. The potential to provide support for exploration that discourages this can be considered. This could include techniques such as systematically reviewing decisions and their implications from different stakeholder perspectives.

**Visualising** The use of visual sketching is a useful means of implicitly managing and experimenting with design alternatives. This appears to be particularly valuable when considering user activity at the device, as it goes some way towards bringing competing issues together in a single representation. Other low fidelity prototyping methods such as paper prototyping serve a similar purpose.

In general we cannot claim these points to be in any way comprehensive, since the focus of analysis within the workshops has been selective, and similarly there is no claim regarding the adequacy of the dimensions examined.


## 6 Conclusions

We have argued that the early stages of software design have distinctive features that are not supported by a wide range of software engineering practices which focus upon development once a design concept is well established. As such early concept design is a process that is complicated by its openness and the unbounded opportunities and promises associated with it. Establishing sufficient details for design decisions and technical assessment to be grounded is core to exploring the design space. Hence, it is valuable to understand how effective exploration takes place, and thus how productive design arguments and development proceed. This problem has been addressed by endeavoring to apply the cognitive dimensions framework as a means of

analysing how the design space is explored in three software design workshops involving professional developers engaging with the same given design brief.

The cognitive dimensions framework has generated a lot of interest as a tool for analysis and assessment of complex interactive systems. The frameworks, and derivations of it, have predominantly been the instruments to conduct analysis. By contrast, in this work we have employed the framework for assessment of observation data gathered with no specific analytic setting. Our approach has been to develop a rational basis for focusing upon and interpreting specific points in the data from the workshops and thus map observed data to concepts in the framework. Future work in the area would hope to develop this analysis approach further. One particularly interesting direction would be to focus more on the visual representations used to articulate and support the design deliberations.

Reflecting upon this interpretation of the data and what appear to be common features of the analysis as a whole we have been able to develop some recommendations for improving the productiveness of early stage software design. The recommendations while based upon the research and analysis identify features of productive design and development that not uncommon to inter-disciplinary group work. Thus while the use of the cognitive dimension framework in this manner is novel the resulting insights are closely reflect elements of good practice.

# References

1. Petre, M., van der Hoek, A. and Baker, A. Editorial in Design Studies, Volume 31, Issue 6, 533–544 (2010) Special Issue Studying Professional Software Design http://dx.doi.org/10.1016/j.destud.2010.09.001
2. Roast, C., Uruchurtu, E. and Dearden, A. The programming-like-analysis of an innovative media tool. In: Psychology of Programming Interest Group Annual Conference, University of York, (2011).
3. Simon, H. A. The Sciences of the Artificial. Cambridge, MA: MIT Press. (1979)
4. Jackson, M. Representing structure in a software system design in Design Studies Volume 31, Issue 6, 545–566 (2010) Special Issue Studying Professional Software Design DOI:10.1016/j.destud.2010.09.002
5. Vennila Ramalingam and Susan Wiedenbeck. 1997. An empirical study of novice program comprehension in the imperative and object-oriented styles. In Papers presented at the seventh workshop on Empirical studies of programmers (ESP '97), Susan Wiedenbeck and Jean Scholtz (Eds.). ACM, New York, NY, USA, 124-139. DOI=10.1145/266399.266411 http://doi.acm.org/10.1145/266399.266411

6. Khazaei, B. and Roast, C. The influence of formal representation on solution specification in Requirements Engineering 8(1), Springer London, 69-77 (2003) Doi: 10.1007/s00766-002-0146-9

7. Green, T. R. G. & Petre, M. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. J. Visual Languages and Computing, 7, 131-174 (1996)

8. Green, T.R.G., Blandford, A.E., Church, L., Roast, C.R. and Clarke, S. Cognitive dimensions: Achievements, new directions, and open questions in Journal of Visual Languages and Computing 17(4), 328-365 (2006) DOI: 10.1016/j.jvlc.2006.04.004.

9. Clarke, S. & C.Becker (2003). Using the cognitive dimensions framework to measure the usability of a class library. In Proceedings of the First Joint Conference of EASE & PPIG (PPIG 15).

10. Triffitt, E. and Khazaei, B. (2002). A Study of Usability of Z Formalism Based on Cognitive Dimensions. In J. Kuljis, L. Baldwin and R. Scoble (Eds), Proceedings of the Fourteenth Annual Meeting of the Psychology of Programming Interest Group, pp. 15-28

11. Hendry, D. G. and Green, T. R. G. (1994) Creating, comprehending, and explaining spreadsheets: a cognitive interpretation of what discretionary users think of the spreadsheet model. Int. J. Human-Computer Studies, 40(6), 1033-1065.

12. Blackwell, A.F., Green, T.R.G. & Nunn, D.J.E. (2000). Cognitive Dimensions and musical notation systems. Paper presented at ICMC 2000, Berlin: Workshop on Notation and Music Information Retrieval in the Computer Age.

13. Green, T.R.G and Blackwell, A. "Cognitive Dimensions of Information Artefacts: a tutorial" Version 1.2    October 1998. http://www.ndirect.co.uk/~thomas.green/workStuff/Papers/ (1998)