Dimension driven re-design – applying systematic dimensional analysis

Chris Roast
Sheffield Hallam University
Sheffield, S1 1WB, UK.
+44 (0)114 225-5555
c.r.roast@shu.ac.uk

Keywords: PO-V.A Cognitive theories, POP-V.B Research tools

Abstract

This paper describes work focused upon the formal analysis of the evaluative framework offered by cognitive dimensions. The paper introduces a tool for assisting with this analysis (CiDa). Findings from using the tool to analyse a series of example systems are described and discussed in relation to proposed cognitive dimension inter-dependencies.

Exploiting Cognitive Dimensions

Cognitive dimensions have received considerable attention within the human-computer interaction and the psychology of programming. The concepts on which the dimensions are based have received broad interest within a variety of domains where the uptake and complexity of notations and artefacts has been seen as a bottleneck in their effective exploitation, see Blackwell and Bilotta (2000), Blandford and Green (1997), Britton et al (1998), Gilmore (1997), Lavery et al (1996), Green and Blackwell (1998). There are examples of applying the dimensions to various information artefacts, in particular: interactive systems, programming languages (textual and visual), programming paradigms, design notations and specification languages. Despite the appeal and apparent appropriateness of the dimensions their role in informing and predicting alternatives within design has not been explored in detail.

Preliminary work in this area has demonstrated that proposed formal definitions of specific cognitive dimensions have been adequate as a means of accurately describing some of the features of specific dimensions, see Roast 1997, Roast 1998, Siddiqi and Roast 1997. The need to develop more complex and robust formal definitions has motivated the development of the CiDa tool as a means of systematically exploring the dimensions.

CiDa - A Tool for Theory Development

The CiDa tool is designed to support the development of a system-based theory of cognitive dimensions. The theory development supported by the tool is viewed as one in which definitions are posited, examined and refined. Support for such a process plays an important role in examination phase, since automated analysis can provide a consistent mechanism for the comparison between successive refinements and case studies. In addition, the complex nature of some definitions can means that they are not necessarily "self-evident". The CiDa tool enables the consequences of posited definitions to be examined when applied to a target system. Thus the analysis outcome provides a set of "observations" about the target system each of which is implied by the theory. The uniform analysis of a target system can be contrasted with selective analysis common in many formal modelling activities.

Exploring the formalisation of specific cognitive dimensions using CiDa has driven the revision of the proposed definitions, with the following patterns of "refinement":

- The reification of a definition - the identification of a number of differing classes of a single dimension.
- The narrowing of a definition - excluding CiDa outcomes that represent "loop holes" in a definition.

- Qualifying definitions - the consideration of specific theories, has lead to the recognition of implicit assumptions regarding their effective use. For example, recognising that a dimension characterisation is reliant upon a system being fully reachable (i.e. that it is always possible to return to any previously visited state.)

A Summary of CiDa's encoded theory

The interpretations of cognitive dimensions employed by CiDa are best expressed in terms of general target system properties. Within CiDa these properties are expressed in terms of temporal logic and behavioural patterns that can be systematically validated for any given target system.

Below are informal descriptions of some of the dimension definitions developed as a consequence of CiDa's use:

Secondary Notation refers to the ability to employ system features that are not formally relevant to the user's primary goal, as mechanisms for conveying other information that is independent of the primary goal. A classic example is the use of comments in programs.

> **CiDa Def.** Cases of secondary notation are represented in terms of a *primary* user goal and the *alternative goals* that are logically independent of the primary when it is satisfied.

Knock-on viscosity refers to the manner in which a target system may behave in such a way that fulfilling a specific goal disrupts other goals that may be of general relevance to the user.

> **CiDa Def.** An instance of this type of viscosity is represented in terms of the *goal* in focus, a *contextual goal*, and an *action*, such that: (i) it is possible for the context to be satisfied, the focus not satisfied and following the action the focus will be satisfied; (ii) no behaviour of the target system allows - the context to be satisfied, the focus not satisfied and following the action the focus and context will be satisfied. In general, the context and focus are distinct and not mutually exclusive.

The above represents a strong form of knock-on viscosity that identifies more specific target characteristics than those publish in Siddiqi and Roast (1997) and Roast et al (2000). These published versions use "weak" knock-on viscosity that is similar in nature but does not address the satisfaction (or otherwise) of the contextual goal prior to achieving the focus goal.

Premature commitment is concerned with when a target system demands the user to make a decision, or enter a state, that is not contributing to their primary goal. Within CiDa, the notion of a user decision cannot be modelled directly, and therefore instances of premature commitment are defined with respect to the action *and* state property that cannot be avoided in reading the primary goal.

> **CiDa Def.** Premature commitment is characterised in terms of the *primary* goal that is to be achieved and the *action* and "*hurdle*" property that the user is unable to avoid. An instance corresponds to: (i) for every behaviour that satisfies the primary goal and encounters the hurdle in doing so, the action leads to the hurdle being encountered; (ii) in the initial state neither the hurdle or primary goal are satisfied. In addition, system behaviours allow the primary goal to be satisfied, system behaviours allow the hurdle to become true, the primary goal and the hurdle are distinct, and reaching the hurdle does not necessarily satisfy the primary goal.

This definition is similar in character to that presented in Roast (1998) however it is more specific regarding the role of the action in premature commitment. The other types of premature commitment are to weaken the final condition above and/or have the hurdle satisfied in the same state as the primary goal is satisfied.

Using CiDa

In order to analyse a target system using the formal definitions introduced above, the target system has to be encoded for CiDa in terms of states and user initiated transitions. In addition, to encoding the target system, a domain mapping is also required, this maps possible user goals to states of the

target system. This mapping is a key element of modelling the target domain as it defines the association between user goals and system features. User interaction with the target system is modelled in terms of user inputs triggering transitions between states.

CiDa analysis output

CiDa is able analyse an encoded target in terms of the formalised dimension definitions introduced earlier. The tool can be directed to verify specific given instances of dimensions for a target system, or to work exhaustively identifying all possible instances of a dimension that a target system may have. This exhaustive mode of use is of interest, since it provides a uniform systematic analysis that can support the comparative analysis of alternative targets. Hence, for a target system *T* and a dimension *D*, CiDa outputs a set of statements each of which is an instance of *D*'s formal characterisation (instantiated with terms from the target system). For instance, CiDa's output for premature commitment can be represented as a table in which each row represents the two goals and an action, for which the definition of premature commitment is verified:

Primary goal	Premature goal "hurdle"	Action
…	…	…
…	…	…

Each row in itself can also be presented as a verbose statement of the instance of the dimension based upon a template. Hence for premature commitment one possible template is:

> When trying to achieve objective <*primary*> using the input <*action*> the user may be prematurely committed to <*hurdle*>

Similarly for knock-on viscosity and the potential for secondary notation the following templates are used:

> When trying to achieve objective <*focus*> using the input <*action*>, <*context*> will be disrupted.
>
> The objective <*primary*> can be satisfied, while the following can be used to convey additional information: <*alternative goal list*>

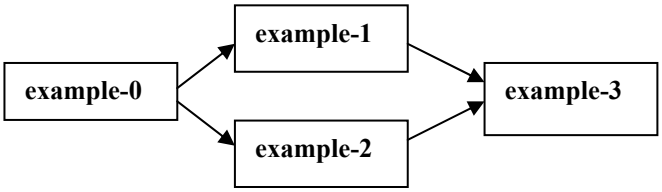
When using CiDa to compare target systems, say *T1* and *T2*, we require that the systems have the same goal mapping and, examine the size and commonality of the analysis of each system, within each of the dimensions. The following section illustrates this use of CiDa in the comparison of alternative example target systems.

Exploration of Designing-out Dimensions

This section describes an exploration of the dimensional analysis provided by CiDa, through taking the example system and modifying it in an attempt to "design-out" specific instances of various dimensions. Using CiDa it is possible to systematically explore how the overall dimension profile of a system is influenced by such modifications. In terms of cognitive dimensions as an evaluative framework, this experimentation supports the examination of trade-offs between dimensions used.

An outline of the exploration of differing target systems:

1. **example-0** is analysed using CiDa and generates instances of premature commitment and knock-on viscosity
2. **example-1** is a re-design of **example-0** to eliminate a specific instance of viscosity.
3. **example-2** is a re-design of **example-0** to eliminate a specific instance of premature commitment.
4. **example-3** is a design conjoining the two modified systems **example-1** and **example-2**.



There are various methodological hurdles to this form of exploration; in particular, the act of "designing-out" a specific instance of a dimension is not deterministic, or obvious. Numerous modifications could be made to enhance and modify a system, and each may have its own influence upon the general profile. The aim of exploring these examples was to consider modifications that have minimal unnecessary impact – thus, the following points were used to focus upon realistic modifications:

- Modifications should be achieved by: (i) the minimal weakening of action pre-conditions, (ii) corresponding changes to post conditions, and (iii) ensuring the corresponding states are legal configurations.
- Modification should not alter the system state structure
- Modifications should not introduce or eliminate actions
- Modifications should not alter the primary purpose of any action
- Modifications should not alter the modelled goal mapping of the target system

Although the exploration of the examples is described in terms of re-design and modification, it is of interest to note that with the formally modelled target systems re-designing and modification can be interpreted in terms of combining and separating sub-systems. Thus, each of the modifications made to a system can be treated as either the introduction or exclusion of behaviours that can be characterised by other systems. Thus, for each modification (from *A* to *A′*), a system *B* can be identified such that $A+B = A'$, and each of the systems (*A, A′* and *B*) can be systematically compared using CiDa. This compositional view of the re-designs is of benefit when we look towards characterising cognitive dimensions more generally.

The initial example system

The initial system **example-0** is based on a module-orientated development environment in which users engage in producing code modules consisting of a name, an import list, an export list and a module body. User inputs allow the introduction of each of these, and the target system partially determines the orders in which this can be done. Hence, the example system embodies a development approach - "encouraging" some development patterns and "discouraging" others. Each of the examples considered operates in the same domain but in effect offers different development opportunities to the user.

The CiDa analysis of **example-0** is summarised below in terms of the number of instances of the dimensions found and, in addition, the number of potential alternative goals (i.e. the goal space) of the example:

System	#goal space	#premature commitment	#knock-on viscosity	Secondary notation potential
example-0	8	4	28	56

Modification to viscosity (**example-0** to **example-1**)

The first modification to the initial example system was to eliminate an instance of knock-on viscosity. In terms of the example domain, there is an instance of knock-on viscosity where by a module cannot be named without it being a stub module. In eliminating this instance the modified system should allow users to provide a module's content and import list without pre-defining the name for that module. This would enable the generation of module content within un-named modules.

The resulting system is **example-1**, the case of knock-on viscosity was eliminated as planned, however two new instances of viscosity were in fact introduced by the modification. In addition, a "side-effect" of the modification has been to remove *all* cases of premature commitment and to increase the potential for secondary notations. This effect can be seen in table 1 which gives the number of instances of differing dimensions for each of the five example systems examined in this paper.

These effects not only demonstrate a between-dimension trade-off in effect, but also an apparent *within-dimension* effect for knock-on viscosity. To explore this within-dimension influence further, a second unplanned revision was made to the example system to see the impact of designing-out one of the new cases of viscosity. The new system is labelled **example-1-1**.

System	#goal space	#premature commitment	#knock-on viscosity	Secondary notation potential
example-0	8	4	28	56
example-1	9	0	29 (28 –1 + 2)	65
example-1-1	10	0	27 (29 – 2)	67
example-2	9	3	32	70
example-3	11	0	31	81

Table 1. Number of dimension instances for the five example systems

In this second modification, the number of overall knock-on viscosity instances was reduced, and there were no new instances of knock-on viscosity created. Collectively this indicates that the modifications can be made to alleviate specific instances of viscosity, however modifications can affect other instances. By contrast, for both the modifications made the measure of secondary notation has grown. The only corresponding attribute is the number of possible goal combinations allowed by the new system. The goals space is a measure of the range of alternative goal combinations that a specific system will allow users to access. Thus it is not un-expected to find that the larger goal spaces provide for more powerful secondary notations.

Modification to premature commitment (**example-0** to **example-2**)

This modification to the initial system was motivated by trying to design-out a specific instance of premature commitment. Despite the fact that one side effect of eliminating a case of knock-on viscosity was to remove all cases of premature commitment, this modification focused upon a single instance of premature commitment.

Within the initial system, in order to define a module's interface the system forces the user to initially create a stub module. The elimination of this instance involved weakening precondition of the operation involved, and ensuring the set of legal states allow the alternative uses of that operation. The result of the modification was to remove the one premature commitment instance focused upon. A side effect of this was to in fact introduce more cases of knock-on viscosity and also increase the potential for secondary notation.

Combination (**example-1-1** and **example-2** to **example-3**)

The final modification considered is that of combining the two examples from above. Specifically, we combine the modifications that eliminated: two cases of knock-on viscosity (**example-1-1**); and one instance of premature commitment (**example-2**), giving **example-3**. Combination was one in which the state transitions are the union of the transitions belonging to the two systems.

In this final case, the goal space has grown and correspondingly, so has the instances of secondary notation. The elimination of premature commitment in the modifications from 0 to 1, has persisted, in examples 1-1 and 3. In contrast, the cases of knock-on viscosity have increased, despite the intermediate example 1-1 having a lower number of instances.

In summary, there are few clear co-relations between the dimension profiles for each example system, other than the potential for secondary notation increasing with the size of the goal space (see table 1). However, it is of interest to note that we've a concrete example of within-dimension trade-off for knock-on viscosity (**example-1** and **example-3**), although the pattern is not consistent (**example-1-1**).

Analysis based on Combinations

To explore the inter-relationship between the instances of specific dimensions when combining systems, the commonality of specific instances can be visualised using a Venn diagram. Figure 1 shows common and distinct instances of knock-on viscosity in the combination of **example-1-1** and **example-2** to **example-3.** Three instances of viscosity within **example-2** were excluded by the combination with **example-1-1**, 25 were common to all.

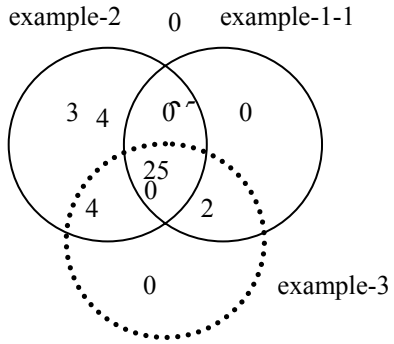


Figure 1. The knock-on viscosity instances between 1-1, 2 and 3

Similarly, the varying potential for secondary notation can be illustrated with the same example systems (figure 2). Overall, 56 were common to the examples and they appear to be cumulative, i.e. none were excluded by the combination.

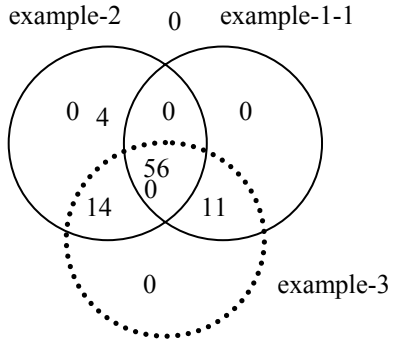


Figure 2. The distribution of secondary notation between 1-1, 2 and 3

Combinations in General

In general all the modifications explored in the examples can be viewed in terms of system compositions. For each modification, we can identify an additional system that, when conjoined with one of the examples gives the same effect as the modification. Thus we can identify and define the following systems:

- **delta-0** - a distinct system capable of achieving the same goals as **example-0**, with the characteristics that in combination with **example-0** it is equivalent to **example-1**.

- **delta-1** - a distinct system capable of achieving the same goals as **example-1**, with the characteristics that in combination with **example-1** it is equivalent to **example-1-1**.
- **delta-2** - a distinct system capable of achieving the same goals as **example-0**, with the characteristics that in combination with **example-0** it is equivalent to **example-2**.

These additional systems and their combinations are open to analysis using CiDa, and can provide further illustrations of how the dimensions measured combine. Figure 3, illustrates the general symmetric characteristics of the analyses for all the above system combinations. (The zero's in figure 3, show where dimensions instances never appear in the examples considered.)

Specific observations arising from the analysis of these combination additional examples are:
- **Non-generative in knock-on viscosity and premature commitment** - In all cases, but for one, the combination of systems creates no more instances of a given dimension than those evident in the systems combined.
- **Generative Secondary Notation -** In one case there is an instance of secondary notation that is generated specifically by the combination, the instance was not evident in either of combined systems.
- **Persistent Secondary Notation** - In all cases, secondary notation persists in the combined system.
- **Non-persistence in knock-on viscosity and premature commitment** - The cases provide three examples of knock-on viscosity and premature commitment being eliminated in the combination.

Discussion

Working with the formal definitions employed by CiDa, has enabled the systematic analyse a number of closely related systems in order examine the dynamics of the formal measures associated with specific dimensions. The general observations drawn from the example analyses suggest various dimension characteristics under composition. These can be compared with the inter-dimension influences suggested by expert analysis of other example sets (Green and Blackwell 1998, Britton and Kuta 2001). Within expert analyses trade-offs between viscosity and both secondary notation and premature commitment are identified. Our small example systems demonstrate one of these clearly, where knock-viscosity was reduced so was premature commitment. However the converse effect is not evident – focusing upon reducing premature commitment has not reduced cases of knock-on viscosity.

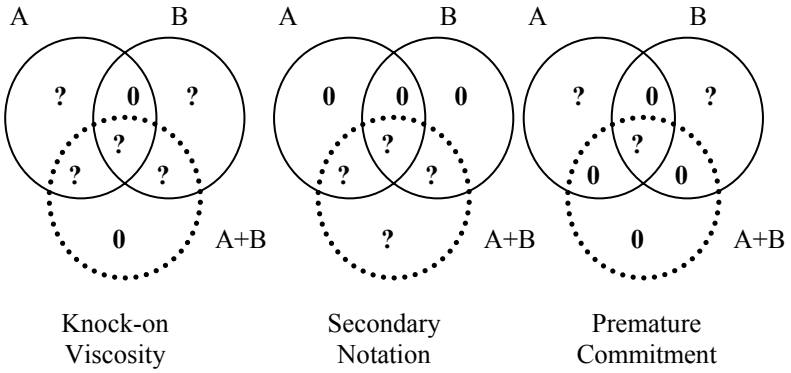


Figure 3. The general features of system combinations for: knock-on viscosity, potential for secondary notation and premature commitment

Although not strongly evidenced in the examples, it is possible to also map the potential relationship between viscosity and potential for secondary notation. Reducing knock-on viscosity on the whole reduces constraints upon the manner in which tasks can be achieved, and thus reductions will only ever increase the goal space (if at all). With CiDa's definition goal space appears to be strongly related to the potential for secondary notation, and thus reducing knock-on viscosity can only increase the potential for secondary notation. The same argument applies for when an instance of premature commitment is eliminated.

Thus, the examples used, and their systematic analysis, appear to conform to the trade-offs suggested in existing work on cognitive dimensions.

Conclusions

The CiDa tool has been introduced and the outcomes of its application to a set of related examples systems has been summarised. The tool provides a systematic analysis suitable for consistent comparison of systems, and from the example set of systems we've been able examine the manner in which dimension instances are altered under system composition.

The future use and development of CiDa is to further explore the widespread validation of the theories used and examine their use in formal modelling. One approach to validation being explored is to relate the CiDa analysis to non-model based assessments of the dimensions, this approach may benefit from the focused development of empirical assessment materials, such as questionnaires (Blackwell and Green 2001, Tukiainen 2001).

References

Blackwell, A.F. & Bilotta, E., Eds. (2000) Proceedings of the Twelfth Annual Meeting of the Psychology of Programming Interest Group. Cosenza, Italy: Edizioni Memoria.

Blackwell, A.F. & Green, T.R.G. (2000). A Cognitive Dimensions questionnaire optimised for users. In Blackwell A.F. and Bilotta E. (eds.), Proceedings of the Twelth Annual Meeting of the Psychology of Programming Interest Group, pp 137-152.

Blandford, A. and Green, T. (1997). OSM an ontology-based approach to usability engineering. In Representations in Interactive Software Development. Workshop at Queen Mary and Westfield College, Department of Computer Science.

Britton, C and Kuta, M (2001) Cognitive Dimensions Profiles: A Cautionary Tale, In Kadoda G. (ed.), Proceedings of the 13th Annual Workshop of the Psychology of Programming Interest Group, pp 265-274.

Britton, C., Jones S., and Lam W. (1998) Separating the System Interface from its Internal State: an Alternative Structure for Z Specifications. Formal Aspects of Human Computer Interaction Workshop, September 1998, Sheffield.

Gilmore, D. J. (1997). Cognitive dimensions as a tool for comparative evaluation. Technical report, Psychology Department, University of Nottingham.

Green, T.R.G. & Blackwell, A.F. (1998). Design for usability using Cognitive Dimensions. Tutorial session at British Computer Society conference on Human-Computer Interaction HCI'98.

Lavery, D., Cockton, G., and Atkinson, M. (1996). Cognitive dimensions: Usability evaluation materials. Technical report, Department of Computing Science, University of Glasgow.

Roast, C. R. (1997). Formally comparing and informing design notations. In Thimbleby, H., O'Conaill, B. and Thomas, P. (eds.), People and Computers XII, pp 315-336, Springer Verlag.

Roast, C. R. (1998). Modelling Unwarranted Commitment in Information Artefacts. In Chatty S. and Dewan P. (eds.), Engineering for Human-Computer Interaction, pp 77-90, Kulwer Academic Press

Roast, C., Khazaei, B. and Siddiqi, J.(2000) Formal comparison of program modification. In Stasko J. and Pfeiffer J. J. (eds.), 2000 IEEE Symposium on Visual Languages, pp 165-171. IEEE Computer Society.

Siddiqi, J. I. and Roast, C. R. (1997). Viscosity as a metaphor for measuring modifiability. IEE Proceedings - Software Engineering, 144(4):215-223.

Tukiainen, M.(2001) Evaluation of the Cognitive Dimensions Questionnaire and Some Thoughts about the Cognitive Dimensions of Spreadsheet Calculation, In Kadoda G. (ed.), Proceedings of the 13th Annual Workshop of the Psychology of Programming Interest Group, pp 291-303.