Integrating Human Abilities With The Power Of Automated Scheduling Systems: Representational Epistemological Interface Design

Peter C-H. Cheng[1], Rossano Barone[1,2], Samad Ahmadi[1,2], Peter I. Cowling[3]

[1]School of Psychology, University of Nottingham, Nottingham, UK
[2]School of Computer Science and Information Technology, University of Nottingham, Nottingham, UK
[3]Department of Computing, University of Bradford, Bradford, UK

email: peter.cheng@nottingham.ac.uk, rb@psychology.Nottingham.ac.uk
sza@cs.nottingham.ac.uk, Peter.Cowling@scm.brad.ac.uk

Abstract

An approach is considered to the integration of the flexibility and knowledge of humans with the computational power of automated systems, in the solution of demanding tasks. The Representational Epistemological (REEP) interface design approach advocates designing representational systems that directly encode the structure of the knowledge of the task domain to serve as user interfaces. A case study is presented in which a novel interface for examination scheduling was design and evaluated, which demonstrates the potential utility of the approach.

Introduction

Automated systems can handle vast amounts of data and exploit sheer computational power to rapidly process information. However, it is difficult to design such systems to use diverse and detailed knowledge of a task domain or to be creative in their solutions to problems. Human experts have diverse and detailed knowledge that they can apply to reason flexibly about a task, and when suitable knowledge is not available they can use heuristics to solve problems. However, human cognitive makeup means that they are severely limited in the speed and amount of information they can handle. How can automated systems and humans be integrated so that the strengths of each one will compensate for the other's weaknesses?

In the Representational Design Principles to Humanize Automated Scheduling Systems Project, we are exploring an approach to address this question. The basis of the Representational Epistemological (REEP) interface design approach is to consider system design, and in particular user-interface design, at an epistemic level; that is using an understanding of the nature of the knowledge of the task domain to constrain and guide system design. There are two aspects to the approach. First, there is the consideration of the conceptual structures and dimensions that task domain experts use and providing the means that such knowledge can be effectively brought to bear on the task. Such expert knowledge includes classes of concepts such as the underlying invariant relations or laws of the domain, symmetries and characteristic patterns, typical and extreme cases, and alternative perspectives.

Second, it is well known in Cognitive Science that the external representational systems that are used for problem solving will dramatically affect the ease of solution, by over an order of magnitude in some cases, and can substantially determine how comprehensible and learnable a domain will be [4, 11]. Representational systems are generative systems that involve the manipulation of symbolic expressions to make inferences (e.g., algebra, geometry, calculus). They are more than mere displays of information (e.g., lists, tables and charts).

The REEP interface design approach combines both of these aspects in the design of representational systems to serve as interfaces that make the "meaning" of the domain transparent and that support efficient problem solving procedures.

REEP interface design differs from other approaches to interface design (e.g., [10]), and information visualization (e.g., [2]), which focus upon finding appropriate combinations of graphical/visual dimensions to present the informational dimensions of a domain. Real task domains characteristically have large multidimensional spaces of information. Arguably, the generic approach in human-computer interaction research is to augment the physical limitations of conventional 2D screen displays by increasing the effective dimensionality of the interface. This may be achieved in many ways, some examples include: graphical dimensions (e.g., color, icons), screen layout (e.g., multiple windows), the temporal dimension (e.g., animation), the (apparent) third spatial dimension (e.g., virtual reality), or by using other modalities (e.g., sound, touch).

In contrast epistemic interface design attempts to mirror the conceptual structure of the domain in the structure of its underlying representation. The conceptual structure of a domain provides a good way to exploit sophisticated geometric, spatial, topological and other graphical schemes to efficiently organize informational dimensions into meaningful patterns with respect to the task domain. By exploiting the invariant relations, symmetries, categories and canonical cases of a domain, epistemic interface design is in effect reducing the number of information dimensions, by putting them in to meaningful configurations, so that they do not have to be individually represented using separate graphical/notational dimensions.

One aim of the present project is to evaluate and extend this approach to the design of representational systems that will integrate the abilities of humans and the power of automated systems. This paper presents a case study of the application of the approach, which involves the design and evaluation of a new diagrammatic representation/interface for a demanding information intensive task that is typically performed by automated systems - examination scheduling.

Automated Examination Scheduling

Examination scheduling is a complex knowledge intensive task that deals with the allocation of exams to time periods and rooms under given constraints. Hard constraints correspond to unbreakable rules (e.g., ensuring that a student is only required at a single place at any given time, room size limits are respected). Soft constraints correspond to preferences. The goal is, whilst satisfying hard constraints, to minimize the degree to which soft constraints are violated (e.g., limiting the number of consecutive examinations for each student, spatial and temporal preferences for some examinations, minimizing the total length of the examination period) [3].

Examination timetables may involve dozens of rooms, hundreds of examinations and thousands of students. Stu-

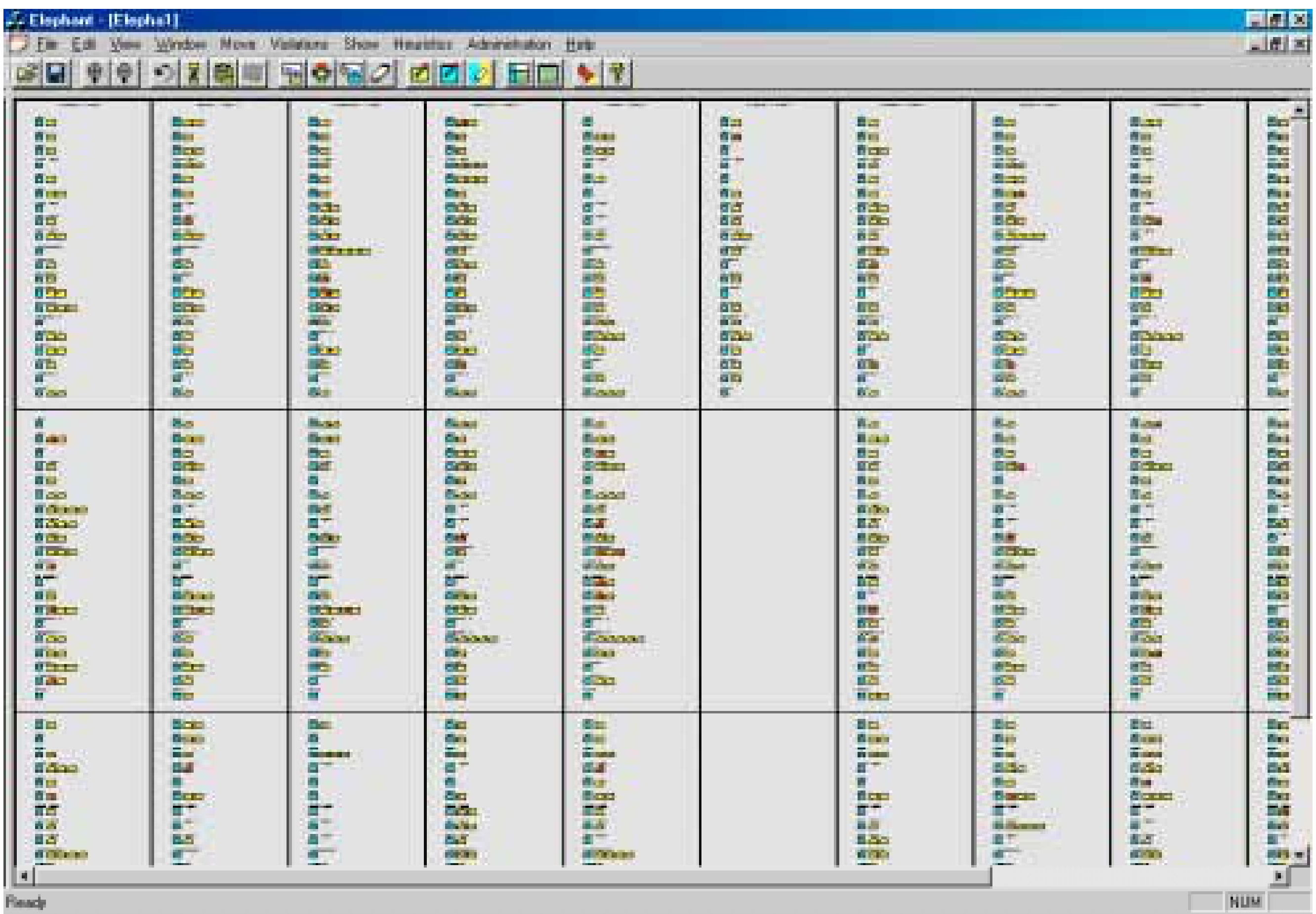

Fig 1(a). A conventional scheduling system interface: global large scale view

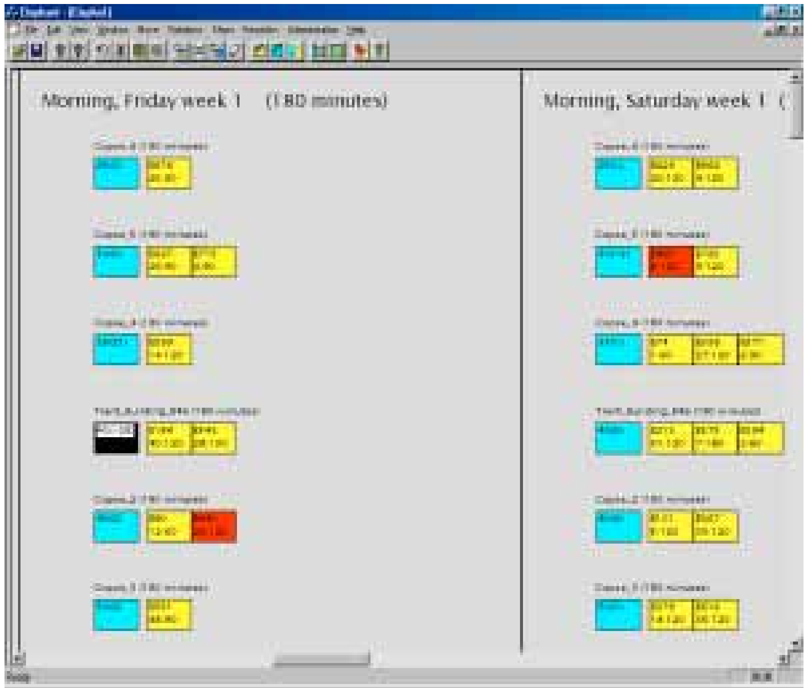

Fig 1(b). A conventional scheduling system interface: local small scale detailed view

dents may take modular courses, allowing an enormous number of possible combinations of examinations for each student. Thus, automated systems for scheduling exams are now commonly used for this demanding task.

Automated approaches use simplified, yet still complex, computer models of the examination scheduling problem. Candidate examination schedules are ranked according to an evaluation function consisting of the weighted sum of constraint violations. Even for these simplified models, creating good timetables can be shown to be computationally difficult in a strict mathematical sense [9]. Automated approaches use constructive and local search heuristics, or meta-heuristics which combine these heuristics using an analogy with other effective problem solving processes, often taken from the natural or physical world, such as natural selection, the organization of crystal structure, or cognitive models of the organization of human short term memory [8]. Usually the design of an effective heuristic for a class of scheduling problems requires substantial human experience and ingenuity, in addition to a good understanding of the generic problem. In any given system the solution heuristics are typically fixed.

The complexities of the examination scheduling problem and the complexity of the automated systems means that users are excluded from the solution generation process. Experienced examination schedulers in different institutions cannot use their specific knowledge of the nature of the local examination problem to change the weights of different factors in the evaluation function, or to choose alternative heuristics that may be more suited to the particular goals and constraints of their institutions. Although they will produce good solutions that satisfy almost all of the given scheduling constraints, the simplified model of schedule requirements used by the automated systems means that parts of their solutions are often unsatisfactory or impractical to implement. In such cases, the scheduler will not normally attempt to manually fix the solutions, but will rather use additional external approaches to overcome the limitations, such as the chaperoning of students who are taking the same exam at different times.

In this respect, our project is addressing two issues. (1) How to make the nature of the methods used by the automated systems understandable, and the operation of the systems accessible to users, so they can use their knowledge to guide the solution process. (2) How to support users' comprehension of solutions (or partial solutions), in all their complexity, so that the solutions can be meaningfully manipulated. The overarching claim is that both these issues can be addressed by designing representations systems to serve as interfaces that directly encode the knowledge of the domain in the structure of the those representations. Here, just the second issue will be considered (see [1] for consideration of the first issue).

Representational Systems for Examination Scheduling

Fig.s 1a and 1b show full screen and detail views of an interface that is similar to the state of the art in commercial automated scheduling systems. It is not an actual interface from real commercial system, but it was implemented to

Fig. 2. STARK diagram representation.

serve as a comparison in the evaluations with the new interface. The basis for its design was to represent selected information about schedules using different graphical dimensions following current approaches to interface design [2]. It uses a tabular spatial substrate that is common to traditional timetables. Time is represented on both axes of the plane. Days are columns and for each day there are three periods shown by three rectangular areas, in each column. In those areas, rooms are the dark blue icons and the exams allocated to the rooms are the light yellow icons. Numbers on the icons show the spatial and temporal information for each exam and room. The room number is written above its icon along with the time it is available. On the room icons, room capacity and current free space (in parentheses) are given. Negative values denote how many students exceed the room capacity. For exams, the number of students taking the exam and the exam's duration are shown. When rooms are colored black, or exams red, this means that the room or exam is involved in some constraint violation, such as room being overcapacity or an exam clashing with an another exam (students allocated two exams at the same time). To see detailed information about the violations the user has to examine additional lists in other windows (not shown).

The conventional representation contrasts with the STARK Diagram (Semantically Transparent Approach Representing Knowledge), which was designed to encode the kinds of knowledge that are needed in scheduling. The knowledge was elicited from project collaborators who are experience examination schedulers and scheduling heuristic designers. A screen shot of the STARK diagram interface is shown in Fig. 2. Three days and nine rooms are shown, but a full size schedule for a typical university can be viewed on a standard computer monitor without undue difficulty.

Fundamental to scheduling is the idea of exams of particular size and duration being allocated to slots, which are rooms of fixed size for particular durations. To support this the diagram integrates magnitude information using a containment metaphor. The spatial substrate has time represented horizontally and space represented vertically, and basic scheduling entities represented by rectangular icons; yellow for exams and blue for slots. Thus, days and periods within each day are represented by columns of slots. The duration of an exam or a slot is the width of its icon.

The size of an exam or the capacity of a room is represented by its icon's height. This representational scheme provides an interpretative scheme for identifying exam and room quantities and also provides a frame of reference for locating and making inferences about the temporal proximity of specific exams, rooms, periods and days.

Further, constraint violation information is integrated into the representation using red lines connecting the sides of the exams. For example, a clash is shown by a line running between exams in the same period and consecutive exams, in which students have one exam immediately following another, are shown by lines between exams in successive periods.

In the empirical evaluations that we have conducted to study how effectively the two representations support the manual improvement of solutions, the STARK diagram is clearly superior, as will be seen below. The STARK diagram has also been incorporated into a prototype scheduling tool (workbench) that allows users to dynamically design heuristics whilst working on solutions to particular schedules. The scheduling tool, called HuSSH is described elsewhere [1, 7].

Representational Design Principles

A major goal of the project is to evaluate, refine and extend a set of principles for the design of representational systems that encode knowledge and support effective problem solving. The principles were used to guide the design of the STARK diagram and have been used to analyze the limitations of the conventional representation. To date, six design principles have been formulated and classified in relation to (1) how the conceptual structure of the domain should be made apparent in the structure of the representation, semantic transparency, or (2) the form of operators and procedures that make for efficient problem solving, syntactic plasticity [5].

There are three semantic transparency principles. (1) Levels of abstraction should be integrated in a single representational system to reduce the conceptual gulf between (a) the overarching laws that govern a domain and (b) specific cases or instance at a concrete level. (2) Alternative perspectives and ontologies should be integrated to allow the alternative perspectives to act as mutual contexts for each other's interpretation. (3) An interpretative framework that combines a globally homogenous with a locally heterogeneous representation of concepts should be provided. It should give a coherent overarching interpretative scheme based on the principal conceptual dimensions of the domain and also have local representational features that make specific conceptual distinctions clear.

There are three syntactic plasticity principles that consider how to make problem solving within a representation easier. (1) The expressions of a representation should not be too rigid nor too fluid, that is they should support malleable expressions, so that all reasonably meaningful expressions can be generated, whilst limiting the generation of arbitrary but otherwise syntactically legal expressions. (2) The procedures for solving problems in a representation

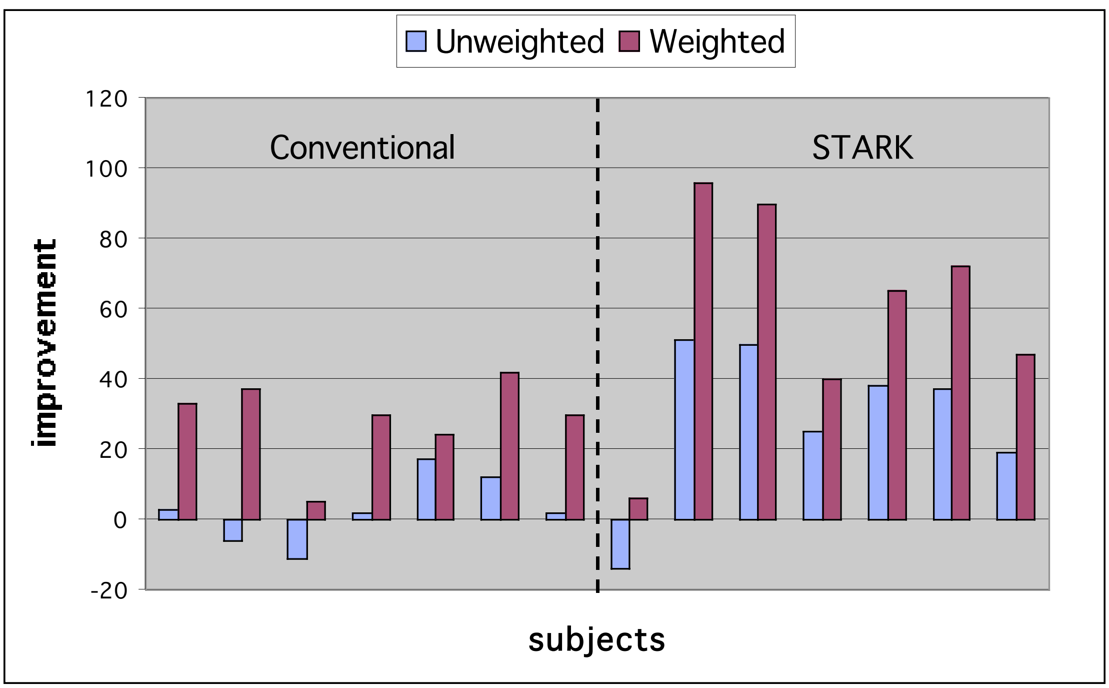

Fig. 3. Performance of users using the two interfaces to manually improve an automatically generated examination schedule. Pairs of bars represent the two measures of performance for each user in each group.

should be compact to minimize the number of operations needed to generate solutions. (3) A representation should have a small variety of consistent and uniform operators making up its problem solving procedures to minimize the complexity of problem solving.

The principles are described more fully elsewhere [4, 5], and more details of how the STARK diagram satisfies the principles, and how the conventional interface largely fails to meet them, can be found in [6].

Evaluation

Users have been trained to manually improve examination schedules with either the STARK diagram or conventional interface. Fig. 3 shows how seven users of each interface performed in the manual improvement of an automatically generated solution to a real university examination schedule, which had 800 exams, 640 slots and over 10,000 instances of exam intersections. Numbers of constraint violations removed (unweighted scores) and violations according to a given weighting scheme are shown. The users were told of the weighting scheme. They were graduate students, who were not experience schedulers, worked for about an hour. The STARK diagram users clearly performed better on both measures.

Inspection of the solution strategies showed that the conventional group engaged in much trail and error search for appropriate allocations. In contrast, the STARK diagrams users appear to have engaged in more deliberated search for problematic exams and suitable slots to which to reallocate them. Further, some of the STARK diagram users adopted recursive strategies in which exams possessing intersections with problematic exams were first reallocated, even though they were not themselves involved in constraint violations, so that clear periods could be created for the problematic exam. Further details of the evaluations can be found in [6].

Discussion

The contrast between the support that the STARK diagram and conventional scheduling interface give to manual solution improvement is stark. The benefits of the STARK diagram can be attributed to the nature of its design, which are largely consistent with the principles of representational design. The purpose of the principles is to achieve mappings so that the representational schemes directly capture conceptual structure of the task domain, making knowledge of the task domain transparent and providing efficient problem solving procedures. Using such representational systems may support the integration of human abilities with the capabilities of automated systems, at least in the case of examination scheduling.

There are, at least, three ways in which effective representational systems may enhance such integration. First, an effective representation can support users' ease of comprehension of the task states generated by an automated system. For example, with the STARK diagram it is relatively easy to see not only overall distribution allocations but also what capacity is remaining and where the constraint violations are densest.

Second, good comprehension of successive task states may enhance users' understanding what processes are being performed by the automated system over time, or even if the automated system is not functioning as expected. For example, we have discovered bugs in the implementation of automated scheduling heuristics by inspecting successive solution states (e.g., biased allocation when random allocation was expected).

Third, making readily apparent the operation of heuristics in the overall context of the structure of the problem provides an opportunity for the design of new heuristics. From example, new heuristics have been suggested by the use of the STARK diagram within the HuSSH scheduling workbench [1], [7]. Specifically, the combination of constructive heuristics and the underlying representational metaphor of STARK diagrams suggested the novel use of solution space partitioning in combination with incremental solution construction within each subspace.

Acknowledgements

The research was supported by a UK ESRC/EPSRC research grant (L328253012) under the PACCIT programme and by the ESRC through the Centre for Research in Development, Instruction and Training (CREDIT).

References

[1] Ahmadi, S., Barone, R., Burke, E., Cheng, P. C.-H., Cowling, P., & McCollum, B. 2002. Integrating human abilities and automated systems for timetabling: A competition using STARK and HuSSH representations at the PATAT 2002 conference. In E. Burke & P. De Causmaecker (Ed.), *Proceedings of the 4th international conference on the practice and theory of automated timetabling (PATAT)*, 265-273. Gent, Belgium: University of Gent.

[2] Card, S., Mackinlay, J., and Schneiderman, B. eds. 1999. *Readings in Information Visualization*. San Mateo, CA: Morgan Kaufmann.

[3] Carter, M.W. & Laporte, G. Recent developments in Practical Examination Timetabling. 1996. In *Selected papers from PATAT'95*, Springer Lecture Notes in Computer Science vol. 1153, 3-21.

[4] Cheng, P. C.-H. 1999. Unlocking conceptual learning in mathematics and science with effective representational systems. *Computers in Education* 33(2-3):109-130.

[5] Cheng, P. C.-H. 2002. AVOW diagrams enhance conceptual understanding of electricity: Principles of effec-

tive representational systems. *Cognitive Science 26*(6):685-736.

[6] Cheng, P. C-H., Barone, R., Cowling, P. I., & Ahmadi, S. (2002). Opening the information bottleneck in complex scheduling problems with a novel representation: STARK diagrams. In M. Hegarty, B. Meyer, & N. H. Narayanan (Eds.), *Diagrammatic representations and inference: Second International Conference, Diagrams 2002*, 264-278. Berlin: Springer-Verlag.

[7] Cowling, P. I., Ahmadi, S., Cheng, P. C.-H., & Barone, R. 2002. Combining human and machine intelligence to produce effective examination timetables. In L. Wang, K. C. Tan, T. Furuhashi, J.-H. Kim, & X. Yao (Eds.), *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL2002)* Singapore: Nanyang Technological University.

[8] Dowsland, K.A. 1998. Off-the peg or made to measure? Timetabling and Scheduling with SA and TS. In: *Selected papers from PATAT'97*, Springer Lecture Notes in Computer Science vol. 1408:37-52.

[9] Garey, M.R., and Johnson, D.S. 1979. *Computers and intractability – A guide to NP-completeness*. San Francisco, CA: Freeman.

[10] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. 1994. *Human-Computer Interaction*. Wokingham, UK: Addison-Wesley.

[11] Zhang, J. 1997. The nature of external representations in problem solving. Cognitive Science 21(2):179-217.