Representations for Problem Solving: On the Benefits of Integrated Structure

Rossano Barone, Peter C-H Cheng
Department of Informatics, University of Sussex, Falmer, BN1 9RN, UK.
r.barone@sussex.ac.uk, p.c.h.cheng@sussex.ac.uk

Abstract

How should problem-solving representations for complex knowledge domains be designed? Traditional approaches typically address the problem of semantic complexity by designing systems that offer multiple and often heterogonous forms of representation. The REEP approach advocates structure preserving integration of the different classes and perspectives of a domain within a single representation. This paper reports on a novel representational system for nurse rostering that was designed under the REEP approach. An empirical evaluation suggests the kinds of knowledge support provided by the representation and demonstrates that participants prefer fully integrated over selective views of information even though the former increases visual complexity. This knowledge support is explained in terms of more abstract domain independent cognitive benefits that we present as reasons for adopting the REEP approach.

1. Introduction

How should interfaces be designed to support problem solving in semantically rich domains? Traditional approaches typically address this issue by designing interfaces that provide multiple representations, each of which express different perspectives or classes of information. In commercial scheduling applications this is typically the norm. State of the art scheduling systems may provide an array of display options using domain independent representations such as graphs, tables and charts.

One of reason for justifying the use of multiple representational systems is to avoid the problem of visual complexity [1]. Taking a multiple representation approach also makes the design task significantly easier to undertake. Designing semantically complex diagrams is difficult. There are limitations on the number of visual dimensions that can be exploited in a two-dimensional representation [3] and this together with well known limitations of the expressive capabilities of diagrams [10] appears to be reasonable case for designers to consider systems composed of multiple and heterogeneous forms of representation.

The approach that we have termed representational epistemology (REEP) holds a somewhat contrary position. The central aim of this approach is to design "domain specific" problem solving representations that optimally encode the task relevant structure of the domain. We advocate the design of such systems using a single representation that integrates the different classes of information, perspectives and levels of abstraction that characterise a domain. These arguments are concerned exclusively with the representation of a problem situation and not other kinds of information that may be associated with a problem (e.g., learning instructions). Our studies have shown that such representations provide greater cognitive support in learning and problem solving compared to conventional representations that typically fail to sufficiently integrate the structure of a domain. Novel diagrammatic representations have been designed for domains in physics, mathematics and real world scheduling problems [4, 5].

This paper provides a further demonstration of the REEP approach with a prototype graphical interface for the semantically complex domain of nurse rostering named STARK-Roster – a new family member of STARK scheduling interfaces (Semantically Transparent Approach to Representing Knowledge). An evaluation study was conducted to assess the knowledge support provided by the STARK-Roster interface and investigate the trade-off between knowledge support and visual complexity. The knowledge support was assessed through the kinds of problem solving procedures participants adopted. In the discussion this knowledge support is explained in terms of more abstract domain independent cognitive benefits that we present as reasons for adopting a REEP approach to the design of problem solving representations. The next section will briefly outline a number of different ways in which a representation can preserve the structure of its represented domain and provide the reader with a flavour of the kinds of structure preserving issues important in the REEP approach to design.

2. Diagrammatic structure preservation

The kinds of representations that preserve the structure of what they represent are broadly diagrammatic in nature. This is why diagrams have been referred to as homomorphic or analogical representations [3]. Many of the cognitive advantages of diagrams seem to be dependent on the structural similarity they have to the situations they represent [3, 5, 9, 11]. There are many ways in which diagrams preserve structure. This section identifies three different perspectives of structure preservation that we have termed: (1) instance structure, (2) dependency structure; and, (3) derivation structure.

One of the most common traits of diagrams is that they instantiate their representing referents (i.e. objects, properties, relations) such that any instance being represented is represented by a single diagrammatic instance. Diagrams also preserve referential connectivity between individual instances of objects, their properties and relations. This pattern of references between the set of represented instances is a form of structure preserved in diagrams.

We use the term dependency structure to refer to systems of interdependent relations. Broadly speaking there appears to be three main classes of dependency structures that may be preserved in diagrammatic representations: taxonomic, logical and arithmetic dependencies.

The capacity of diagrams to encode class relations, have been noted by a number of authors [3,5]. The REEP approach has emphasised the role of designing representations that organise represented objects, properties and relations according to over-arching classes or dimensions of the knowledge domain. We refer to such interpretive schemes as globally homogeneous [2, 4, 5].

Another class of complex dependency structures that may be preserved in a representation are logical dependencies. Consider temporal relations for instance (e.g., before/after). These relations are transitive, anti-symmetrical and anti-reflexive. Diagrammatic systems that have these properties (e.g., left of/right of) will preserve this logical structure. Arithmetic dependencies of a represented situation can also be preserved in a diagrammatic representation through geometrical configurations. Preserving logical/arithmetic structure in a diagrammatic system ensures that it is incapable of generating logical/arithmetic inconsistent expressions [4, 9]. Cheng refers to such classes of diagrams as Law encoding because these diagrammatic constraints have been used to express principal laws that govern a represented system [4].

Given a specification of the mappings of the primitive represented objects, properties and relations in a representation, emergent expressions representing higher-order relations can be derived from diagrams. Cognitively speaking this is a case of selectively attending to the representing referents and appears to be supported to the extent that the expressions exploit mechanisms of perceptual organisation. Shimojima refers to these phenomena as derivative meaning [8]. The pattern of relations between the represented things that enter into a higher-order derivative expressions have a particular structure themselves. Following Palmer we use the term derivate structure to refer to the composite structure of particular expressions [7]. The derivative structures of an expression may have different degrees of correspondence to the way the same meaning is derived in the represented situation. Cheng has demonstrated that complex diagrammatic representation can be designed to simultaneously express alternative perspectives or different levels of abstraction of a domain through interdependent derivative structures. Research under the REEP approach suggests that integrating alternative perspectives and levels of abstraction within a single representation, provides substantial support to leaning and problem solving.

3. Nurse Rostering

Nurse rostering is a combinatorial optimization problem that requires the assignment of a set of nurses to a set of shifts over the course of some planning period under varied constraints. Nurse rosters may be generated by hand using tabular style representations or though automated rostering systems. As with other scheduling systems we have investigated [2, 3] conventional interfaces represent different perspectives or classes of information through separate windows using domain independent representations such as tables and charts. Whether the scheduler generates the roster from scratch or edits a computer generated solution, these interfaces typically fail to provide good cognitive support.

There are a variety of different constraints that may vary across different organisations. For this research we have considered four main classes of constraints. (1) *Working-hours*. Each nurse has a minimum and maximum number of working hours they may work each week or month. The constraint on working hours is violated if a nurse is assigned too few or too many hours. *(2) Staff-requirements*. For each shift a ward requires a specific number of nurses with the right mix of skills based on attributes such as qualifications, training and rank. Constraints on staff requirements are violated for a particular shift if a ward has been assigned to many or too few staff with the correct mix of skills. (*3) Rest-period.* Nurses must have an adequate period of rest between consecutive working shifts. The constraint is violated if a nurse has not been given sufficient rest-period between assigned consecutive shifts. *(4) Nurse-Preferences.* Nurses specify preferred shifts and preferred days off before the final roster is created. The scheduler must try to match these preferences. Violations of preference constraints occur when nurse preferences have not been met.

4. The STARK-Roster diagram

This section provides a brief overview of the STARK-Roster diagram (see Figure 1). Each shift is represented by burgundy coloured column and each collection of three columns represents a day. Shifts, days and weeks are ordered along the horizontal axes. Nurses are ordered along the vertical axes based on their qualifications and skills. Each horizontal line represents a timeline of a single nurse over the course of the planning period. An assignment of a nurse to a shift is represented by a green rectangle; the location and width of the rectangle expresses the date and duration.

Staff-requirements. In the STARK-roster diagram a set of requirements associated with a particular shift is represented as a collection of columns nested in a shift. Each column of the collection represents a particular requirement condition. When a cell in a requirement column is highlighted the nurse indexed by the vertical position of the cell indicates that the nurse is a member of the set specified by the requirement condition. If a cell is transparent then the corresponding nurse is not a member of the set specified by the requirement condition. A requirement can be over-assigned, under-assigned or satisfied. When a requirement is under-assigned the cell will be shaded white, when satisfied grey and when over-assigned black. Specific numerical values about requirements are currently expressed through the status bar when a requirement condition is selected.

Working hours. The violation states of the nurses working hours are represented in two ways. A short horizontally structure extending beneath the left of the timeline expresses the existing consumption and constraints on working hours. The length of the green line represents what is consumed. If the end of this line falls in the surrounding white region the nurse's hours are under-assigned, if in or on the threshold of the grey region the nurse's hours are satisfactory, if beyond the grey region the nurses hours are over-assigned. Violation states are also represented by the shade of the outer-casing of the nurse timeline. When a nurse is under-assigned the timeline will be shaded white, when satisfied grey and when over-assigned black.

Preference constraints. Nurse preferences to work or to take a day off are represented by plain or crossed icons located on the nurse's timeline. Preferences to work that are not assigned will have their icons shaded white showing that the constraint is under-assigned and grey if assigned showing that the preference is satisfied. Preferences to be off on a day that are assigned will have their icons shaded black expressing the preference is over-assigned and grey if not assigned showing that the preference is satisfied. The height of the preference icon represents how serious this constraint should be considered. Preference icons can also represent more specific types such as annual leave, study days etc. with the addition of iconic symbols located on the preference icons.

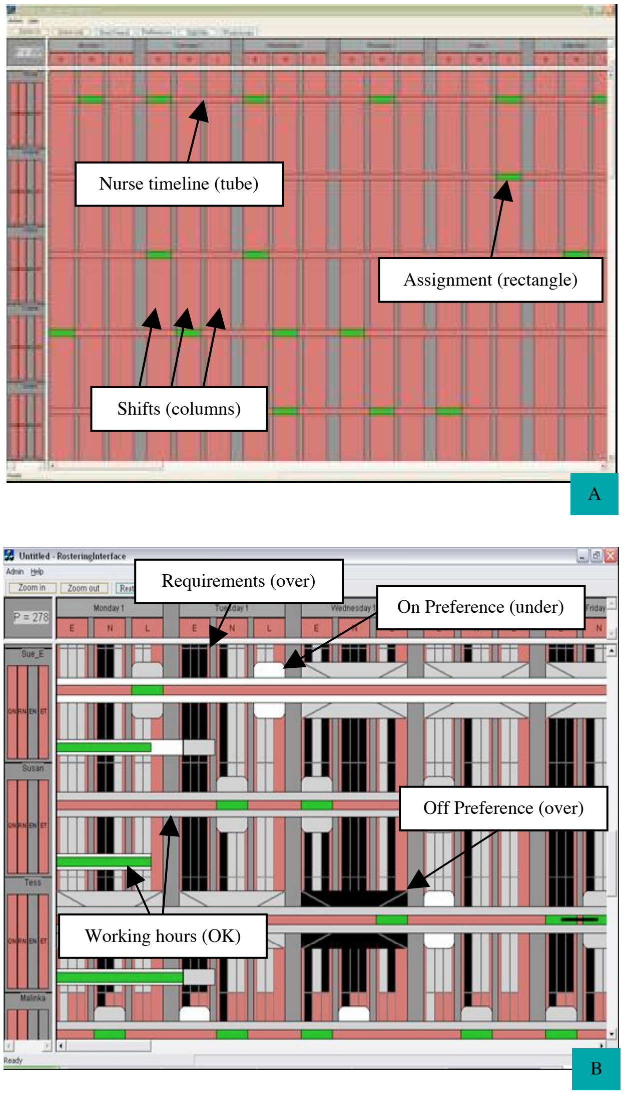

Figure 1. The STARK-Roster Interface with all constraints hidden (A) and all showing (B)

4.1. Global homogenous scheme

One of the important design principles of the REEP approach concerns the structuring of a representation according to the over-arching dimensions or classes of information present in the domain. We use the term global homogeneity to refer to this semantic trait of representations. One example of a globally homogeneous interpretive scheme in the STARK-Roster interface can be observed in the representation of violation states for the different types of constraints. The triple scheme maps white, grey and black to under–assigned, satisfied and over-assigned states for all types of constraints. The referential juxtaposition of the different constraints and their states provide perceptually salient emergent expressions. For example the predominance of white or black constraint icons at a slot will reveal a place where appropriate changes to a slot satisfy the most constraints.

4.2. Problem solving procedures

This section considers basic problem solving procedures for editing schedules: (1) forward-checking, (2) weighted-selection and (3) recursive strategies.

Forward-checking refers to a look-ahead procedure in which the user checks what constraint violations will arise before a change to an assignment is made. Our research suggests that whether a user decides to check dependencies is to some extent determined by the amount of effort required to access the information. If accessing this information is cognitively expensive users may be more inclined to resort to trial and error behaviour [2]. Forward-checking in conventional style interfaces can be difficult as different types of constraint information, if present, is typically represented in one or more different frames of reference.

Weighted-selection refers to a procedure in which the users are able to visually pick out assignments or slots that are causing the most problem. Assignments/slots that have dependent constraints that are largely under-assigned or over-assigned are salient in the STARK-Roster. Changes to these assignments/slots satisfy the most constraints. Weighted-selection allows users to adopt strategies at a more global level of consideration.

Previous research with human scheduling identified exploratory strategies for rescheduling which we refer to as recursive strategies [2, 4]. In recursive strategies, users make a sequence of changes to a solution to solve some goal that involves one or a set of interconnected constraint violations. Recursive strategies occur when constraint violations cannot be resolved by a single step. A simple example of this in nurse rostering may occur when a preference to work is violated but assigning the nurse to the preference would cause a staff requirement violation. A recursive strategy would involve making this assignment with a view to removing a different assignment to resolve the requirement violation.

5. Method

A pilot evaluation was conducted for two reasons: (1) to establish the level of knowledge support provided by the STARK-Roster as measured by the use of problem solving procedures such as weighted-selection and recursive strategies and (2) to assess whether information complexity would be preferred at the cost of visual complexity as measured by the use of tools to hide or reveal selected classes of information.

5.1. Design

A total of six postgraduate students of the Department of Informatics at the University of Sussex took part in the evaluation. None of the participants had previous experience with the nurse rostering problem. All participants were paid for their involvement

The evaluation task required participants to manually improve a roster by reducing violations of constraints. In addition to the main diagram the interface toolbar also had zoom controls to view the STARK-Roster at different levels of granularity. There were also four view controls each of which corresponded to one of the four different classes of constraints. The view controls allowed the users to hide any combination of these constraints. There were three types of edit operations for changing the STARK-Roster. Participants could add assignments, remove assignments and switch assignments from one slot to another (note that switch is composed of a remove and an add operation). All edit operations were generated through simple mouse actions

The novice participants first completed a short tutorial on the rostering problem and interface. This was followed by a twenty minute practice session at improving a roster before the evaluation session began. In the evaluation session participants were given one hour to improve a roster to the best of their ability and were provided with a simple weighting scheme for the constraints. The weighting scheme was: one point per shift a nurse was over or under-assigned (working-hours); one point per nurse that was over or under-assigned to a requirement condition; one point per preference violation; one point per instance of a rest-period violation. Interface operations were logged throughout the evaluation session.

The two nurse rosters used in the practice and evaluation sessions were modifications of real data sets taken from a hospital. The modifications were made to resolve various inconsistencies and to simplify the problem that the novice participants had to learn. The problem involved a roster with nineteen nurses and planning period of twenty-eight days. There were three shifts in each day.

5.2 Results

This section discusses the results of the evaluation task. The percentage of weighted violations reduced by participants was substantial and variability between subjects performance was small (M=61%, SD=4) suggesting that participants' strategies were effective and consistent between subjects. A large number of edit operations were made over the one hour evaluation session (M= 496, SD =133) which amounts to an average of one operation approximately every seven seconds.

The zoom and view controls were analysed to asses whether visual complexity was perceived as a problem by participants. Log files reveal that participants did not use the view controls. There was only one instance of hiding a view and this was refreshed a second after. Zoom-controls on the other hand were used by all of the subjects. Two subjects zoomed out to an intermediate scale for the whole of the session, others zoomed in and out although the frequency of using the zoom controls is small (M=4.3. SD=3.1). This data suggests a preference to use global as well as local interpretations of the roster although global perspectives were more common.

Recursive strategies were assessed by extracting the number of cases where the following conditions where true between consecutive edited operations. The first edit operation E_i: (1) improved one or more constraint violations and (2) simultaneously made one or more dependent constraint violations worse. The subsequent edit operation E_i+1 (3) improved a constraint violation made worse by operations E_i and (4) did not involve undoing E_i. Analysis reveals that the mean percentage of edit operations where the stated conditions were true for E_i was substantial (M=24%, SD=5%) suggesting recursive strategies were commonplace amongst all participants.

To assess the amount of weighted-selection we computed for each type of operation the number of occasions that participants added or removed assignments that satisfied one, two or three of the types of constraint violations (excluding rest-period). There is a chance level of making weighted-selections which is dependent on the number of weighted selection opportunities available in a given roster. To estimate the chance level, we computed from the initial roster, the number of assigned slots that had one, two and three types of constraints that were over-unassigned (*add estimate*) and the number of unassigned slots that had one, two or three types of constraints that were under-assigned (*remove estimate*). Note that this is a conservative estimate because it excludes from consideration cases where individual constraint violations require more than one assignment change to be resolved.

Table 1 shows the percentage of weighted-selections for each participant, the group means and baseline estimates. Note the baseline add estimates are greater than the baseline remove estimates revealing a disproportionate number of weighted-selection opportunities for under-assigned slots. One can observe that on average more than twice as many add and switch operations (the add part) involved in reducing under-assigned violations (59%) satisfied two or three types of under-assigned constraints which is more than twice the value of the baseline estimate (25%). The mean percentage is less prominent for remove and switch operations (remove part) where only about 12% of these operations involved reducing two or more types of constraints with over-assignment violations, although this is still greater than the baseline estimate (7%).

Table 1. Percentage of add and remove operations that satisfied one, two or three types of constraint. (Add / remove parts of switch operations included).

Subject	Add (Under)			Remove (Over)		
	3	2	1	3	2	1
1	8	48	44	1	15	84
2	16	37	48	0	12	88
3	8	52	40	0	9	91
4	9	50	41	0	14	86
5	15	51	35	0	8	92
6	16	47	37	0	11	89
Mean	12	47	41	0	12	88
Baseline	2	23	75	0	7	93

6 Structure Preserving Integration

Despite the visual complexity of the STARK-Roster interface, the view controls were not used by participants in the test session. Given the choice of viewing all the classes of information together or selecting tailored views of specific classes of information, participants opted for the former despite the increase in visual complexity. Zoom-controls on the other hand were used by all of the subjects. The data suggests a preference to use global as well as local interpretations of the roster, although global perspectives were more common despite the increase in visual complexity. Participants edit operations also suggests that the representation supports problem solving procedures such as weighted-selection and recursive strategies. These procedures allow rostering to be significantly more efficient but are clearly dependent on the simultaneous presentation of the different classes of information. It seems reasonable to suggest that participants sacrificed any negative costs of visual complexity to gain problem solving efficiency.

At a more abstract level the problem solving advantages implicated by integrated structure preservation can be understood in terms of more generic notions of cognitive support.

In section 2 we discussed the preservation of instance structure. Instantiating representing referents and the connection between them within a single representation supports referential coherence – the capacity of the cognitive system to bind token objects and their relations in derived expressions [6]. When the situation being represented comprises of many different types and instances and the information requirements in problem solving is complex, the impact of this support increases substantially. Having multiple representations, where replications of the same represented instances are made over different presentations of information, makes the task of establishing correspondences between different instances difficult.

The structural form of a representation can determine what knowledge and procedures become activated [2, 11]. As diagrams are analogical representations they have a default interpretation independent of what they represent. The more meaning/structure preserved in the default interpretation of a diagrammatic expression the more influence it should have on knowledge construction. For example, recognizing whether an assignment to resolve some requirement will or will not have effects on the states of other requirements in the same shift is almost unavoidable. This is a result of the set relations present between requirement conditions and nurses.

Conceptual integration involves selectively mapping distinct packets of knowledge from different sources to understand and reason about a situation. The capacity to support it with an external representation is dependent on a number of different forms of structure preservation as discussed in section 2. Generally speaking the richer the structure preservation the greater the kinds of derivative meaning the representation will provide. Both weighted-selection and recursive strategies involve significant conceptual integration. Such strategies would not be well supported under a multiple representation systems, especially one that employs relatively arbitrary forms of representation.

7. Conclusion

The REEP approach advocates the design of representations that integrate the underlying structure of the domain relevant to the task. The paper provided an example of such a representation for the semantically complex domain of nurse rostering. An empirical evaluation reveals the kinds of knowledge support provided by the novel representation and shows that users prefer full integration of classes of information over selective views of information despite the visual complexity associated with the former. These preliminary results provide additional support to the REEP approach to representational design.

Acknowledgements

The research was supported by an ESRC/EPSRC research grant (L328253012) under the PACCIT programme

References

[1] Ainsworth, S. E. (1999). The functions of multiple representations. *Computers & Education*, 33, 131-152.

[2] Barone, R., Cheng, P. C.-H., Ahmadi, S., & Cowling, P. I., (2003). The strategic influence of conceptual structure in graphical interfaces for scheduling. In P. G. T. Healey (Eds.),. *Interactive Graphical Communication Workshop 2003*. Working Papers. Queen Mary: University of London, 7-20.

[3] Barwise, J., & Etchemendy, J. (1995). Heterogonous logic. In: Glasgow, J., Narayanan, N. H., & B. Chandrasekaran, B., (Eds.): *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. AAAI Press Menlo Park, CA, 211-234.

[4] 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*. Berlin: Springer, 264-278.

[5] Cheng, P. C-H. (2002). Electrifying diagrams for learning: principles of complex representational systems. *Cognitive Science*, 26, 685-736.

[6] Larkin, J. H., & Simon, H. A: (1987) Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11, 65-99.

[7] Palmer, S. E.: (1978). Fundamental aspects of cognitive representation. In: Rosch, E., & B. B. Lloyd, B. B. (Eds.), *Cognition and Categorization*. Hillsdale, N.J.: Lawrence Erlbaum, 259-303.

[8] Shimojima, A. (1999). Derivative meaning in graphical representations. *Proceedings of 1999 IEEE Symposium on Visual Languages*, 212-219.

[9] Stenning, K., Inder, R. & Neilson, I. (1995). Applying semantic concepts to analyzing media and modalities. In: Glasgow, J., Narayanan, N. H., & B. Chandrasekaran, B., (Eds.): *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. AAAI Press Menlo Park, CA, 303-338.

[10] Stenning, K. & Oberlander, J. (1995). A cognitive theory of graphical and linguistic reasoning: Logic and implementation. *Cognitive Science*, 95, 97-140.

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