A Representational Construct for Sharing Knowledge in Design Exploration

Sambit Datta
School of Architecture and Building, Deakin University, Australia
http://www.deakin.edu.a/~sdatta

Exploration with formal design systems comprise iterative processes for specifying problems, finding plausible and alternative solutions, judging the validity of solutions relative to problems and reformulating problems and solutions. These processes are knowledge intensive, collaborative and multidisciplinary in nature. Recent research efforts propose representational frameworks that allow for modeling of knowledge capture, knowledge sharing and knowledge reuse during designing. However, design remains a human enterprise: to be scalable and usable in design practice, formal symbolic representations need to be embedded within a broader framework of agent (human and computational) interaction. This paper argues that, for sharing and reusing knowledge between agents in design exploration, it is necessary to build an intermediary representational structure that bridges specialist interactions with exploration knowledge (the domain) and the symbol structures that represent them (the symbol substrate). The paper identifies the requirements of such an intermediary representation for the sharing of knowledge between design agents. These requirements are addressed through the development of a shared interaction construct, the feature node.

Keywords: *exploration, design knowledge, interaction model, mixed-initiative*

Introduction

Supporting computational design exploration requires flexible and extensible representational structures for supporting the iterative process of specifying problems, finding plausible and alternative solutions, judging the validity of solutions relative to problems and reformulating problems and solutions. Recent studies report representational frameworks that allow for flexible feature-based modelling (Leeuwen and Wagter, 1998), sorts (Stouffs and Krishnamurti, 2002) and design exemplars (Summers et al., 2002).

However, design remains a human enterprise: to be scalable, representational formalisms need to be embedded within a broader framework of human computer interaction. To support the user in designing with formal systems, shared representations that integrate the role of the designer and generative formalisms are necessary. For example, an exposition of the designer's actions in design space is captured in the metaphor of navigation (Chien and Flemming, 1996). Navigation encapsulates the user's action in design space through the traversal of paths and landmarks defined over the navigation structure. Flexible representational structures (Stouffs and Cumming, 2003) and the visualization and interaction with such mixed representational data constructs allow users novel mechanisms for interaction. Shared representational structures enable the designer to formulate problems, generate solutions, make choices and visually browse the history of exploration (alternatives, revisions).

Mixed-initiative (Cohen et al. 1998) is one model of interaction proposed for representations that are shared between human and computational agents. Mixed-initiative enables both user and formalism to exercise joint responsibility over domain goals. A graphical notation supporting mixed-initiative human-computer dialogue is reported by Datta and Woodbury (2002). The paper argues that, for sharing and reusing knowledge between agents participating in design exploration, it is necessary to build intermediary representational structures bridging the agent's view of exploration and the symbol structures that represent them.

This paper explains an interaction construct, the feature node, developed to support mixed-initiative design exploration. The feature node as an interaction construct provides a representation that is shared between agents. The requirements for such an interaction construct and its representation in terms are described in the paper.

Requirements

Designing is a reflective conversation that involves the recursive processes of seeing, moving and seeing (Schön and Wiggins, 1992). Choices, alternatives and versions emerge out of the interaction between designing (acting) and discovering (reflecting). The designer's view of exploration comprises problems, solutions, choices and history (their connections). The problem formulation and reformulation cycle, the solution generation and reuse cycle, the intentional choices made by the designer and the rationale of exploration history need to be captured by the representation. The requirements of the shared representation are as follows:

- Problems and Solution representations must be shared between the designer and the formalism,
- Choices. The intentional choices made by the designer during exploration must be captured by the representation,
- Exploration history. The rationale of exploration from problem definition, reformulation, and solution generation must be captured in the representation.

Problems and solutions co-evolve (Hybs and Gero, 1992) and reformulation is an integral part of the problem definition (Smithers, 2000). Problems and solutions in design are inherently partial (Woodbury et al., 1999). Generated solutions provide a large space of alternatives (Woodbury and Chang, 1995) and representational models can reduce cognitive overload and facilitate choice making (Chien and Flemming, 1997). Choices, their connections and the developing history of explicitly discovered design alternatives must be accessible to the designer through interaction with the structure of exploration. Exploration rationale (Smithers, 2002) and design history (Burrow and Woodbury, 2001) are significant tools for supporting exploration. The designer must be able to exploit this history through navigation and recombination of the paths of exploration.

The next section describes how the feature node facilitates the sharing of a common representation between agents (user and formalism) in formulating and reformulating problems, generating solutions, making choices and navigating the history of exploration.

Feature Nodes

The feature node is an interface construct for addressing the above requirements, composing the designer's view of exploration and the formal substrate. A feature node, FNode, encapsulates the designer's interaction with the formalism by coupling user actions with the elements of the underlying symbol level. a feature node can be conceptualised as a form of directed graph as depicted in Figure 1.

In Figure 1, node 4 represents a general property class. Two examples of features are shown, where the attributes colour, rvalue of node property point to nodes 5 and 6. The formal representation of the feature node can also be represented visually in attribute-value matrix notation. This representation is shown in Figure 2. By representing this structure in visual form, it is possible to enable the designer to access the elements of a feature node directly.

The feature node connects to the formal substrate and supports problem formulation and reformulation, solution generation, choice-making over problems and solution alternatives and navigation of exploration history. Thus, the feature node, FNode, is a representational construct integrating a designer's view of exploration comprising problems, solutions, choices and history over the symbol level representation of design space exploration. Problem state, Solution state, Choice and Operations are explained through the FNode. The mappings between the above concepts and their relationships are modeled in using the unified modeling language, UML, notation (Jacobson et al., 1998). The feature node models the relationship between a problem state and a partial solution to the problem as shown in Figure 3.

Problem States and Solution states

A problem state corresponds to the designer's view of problem formulation. A solution state corresponds to the initial, intermediate and final designs satisfying a problem. The feature node must capture the connections between a problem and its possible solutions (partial or complete) uncovered in exploration. Problems need not be fixed. Designs can be partial or complete with respect to the initial problem formulation.

Firstly, the feature node, FNode captures the dynamic and changing relationships between a problem state, PState and alternative designs (partial solutions) to the problem, SState. By interaction with this element, the designer can either modify (reformulate) the PState or generate a new problem state. The elements of a FNode are entities representing SState nodes generated by the formalism. These are the partial satisfiers (solution states) of the FNode. By interaction with these elements, the designer can unfold the possible solution states of the current problem.

Intentional choice

Problems and requirements have multiple solutions. In design, a problem formulation may have no solutions, a finite number of solutions or an arbitrarily large collection of solutions. The intentional choices made by the designer in the reformulation of problems and commitment to generated solutions during exploration are recorded in the feature node.

Exploration history

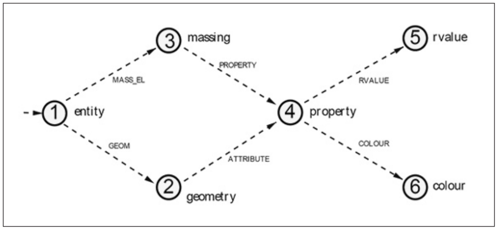
Exploration history captures the rationale of exploration as a record of designer actions, formal moves and choices. This history is captured in the shared representation as a collection of ancestor and progeny feature nodes. This collection of ancestor and progeny feature nodes is termed a satisfier space (to distinguish it from the underlying design space). This satisfier space records problems, solutions and designer choices and therefore encapsulates the rationale of exploration. The relationship between feature node collections in satisfier space is shown in Figure 4.

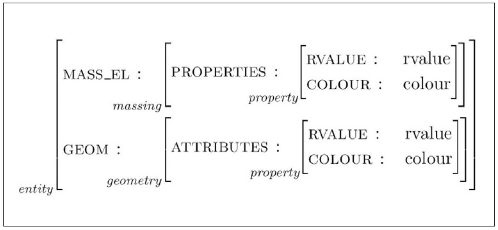
Example implementation

The shared representation presented above has been implemented in a demonstration prototype (Datta, 2004). It uses the example of massing configurations as an example domain and extends the example presented in Woodbury et al (1999). Further details of the implementation and examples

Figure 1, left
The feature node is represented as a directed acyclic graph (DAG).

Figure 2, right
A feature node shown visually in attribute-value matrix notation.





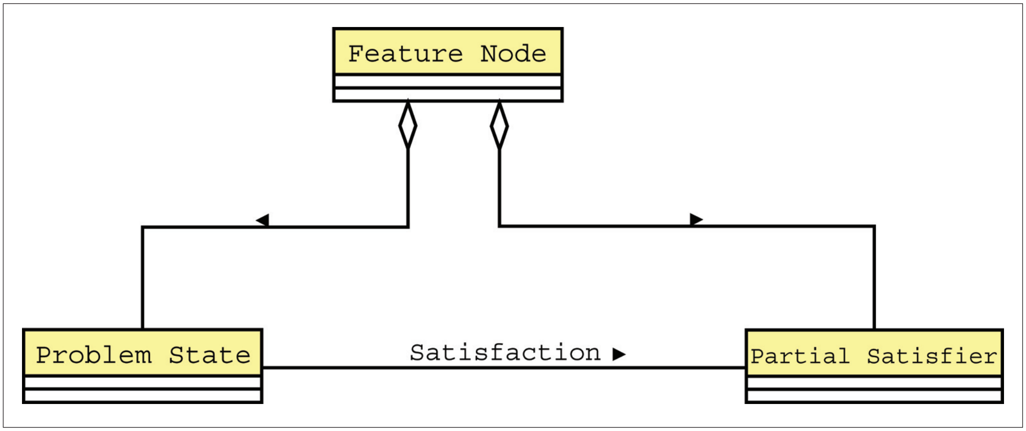


Figure 3
The feature node models the relationship between a problem as a problem state and a partial solution.

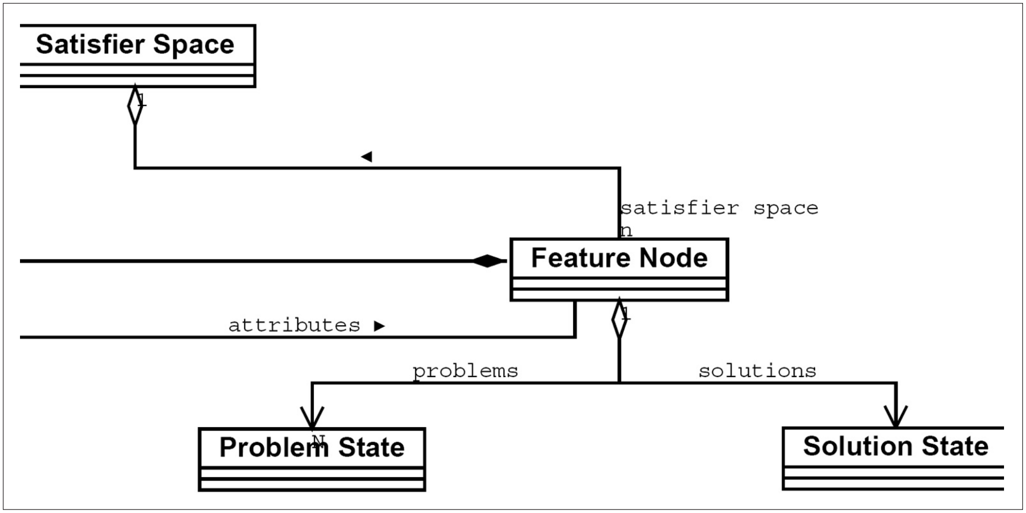


Figure 4
The collection of ancestor and progeny nodes forms a tree of exploration structure. This tree is termed a satisfier space. Through the feature node, problems, solutions and their history are recorded in satisfier space.

under developed are given in (www.deakin.edu.au/-sdatta/folds).

Feature nodes are implemented as elements in satisfier space. In Figure 5, Problem statements are encoded on the left. Solution states are recorded in the right. Exploration history of both problems and solutions are recorded in the representation. The possible future explorations of the selected node are shown as a list of features (bottom right, Figure 5).

The feature node, FNode captures the dynamic and changing relationships between a problem definition and its explored solutions. The problem state, PState and alternative designs (partial solutions) to the problem, SState are recorded in the shared representation. Thus when a problem is modified, the formalism generates a new solution state and inserts it in the appropriate node in the current exploration. When a solution is modified, the modifications are recorded in the shared representation as choices made by the designer.

Discussion

The shared representation explained in this paper supports the requirements for mixed-initiative exploration. The collection of feature nodes, the satisfier space, is a representation of the rationale of exploration. This satisfier space captures problem formulation and reformulation, solution generation, solution reuse, choices made by the designer in a single shared representation. Through a single representational structure, the feature node, it is possible to encapsulate and account for both user

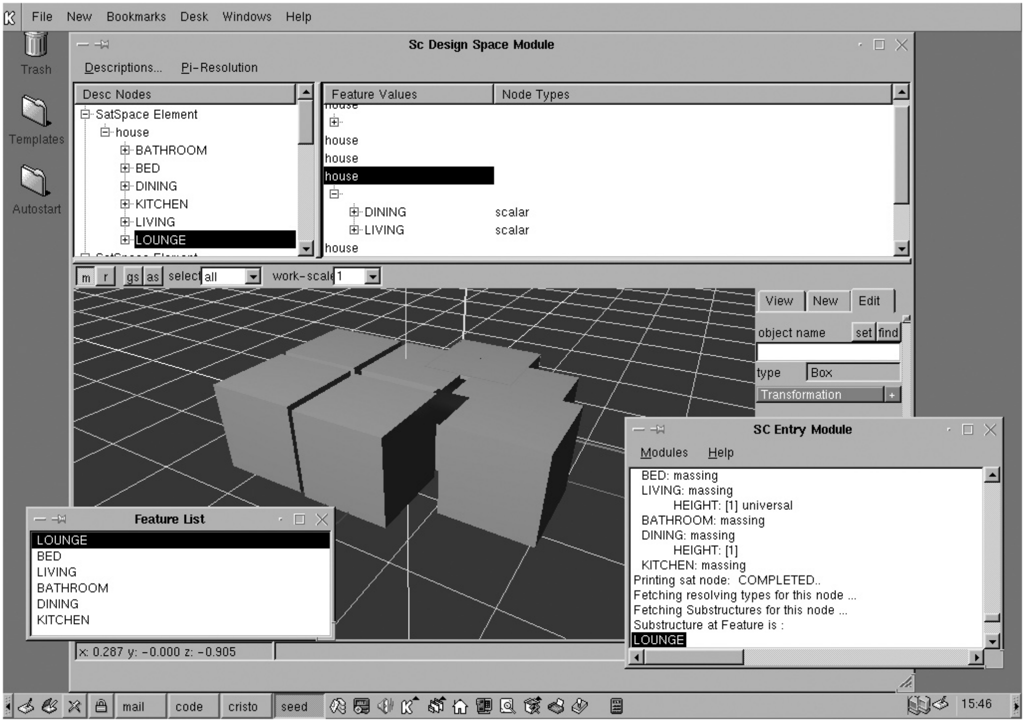


Figure 5
Feature nodes are implemented as elements in satisfier space (top half of window). Problem statements are encoded on the left. Solution states are recorded in the right. Exploration history of both problems and solutions are recorded in the representation. The possible future explorations of the selected node are shown as a list of features (bottom right).

actions and formal moves. Further, this interface construct permits the sharing, capture and reuse of exploration knowledge.

Acknowledgements

The author would like to acknowledge the contributions of Professor Robert Woodbury, Simon Fraser University.

References

Burrow, A. and Woodbury, R.: 2001, Design Spaces–The forgotten Artefact, in M. Burry, T. Dawson, J. Rollo, and S. Datta (eds.), Hand, Eye, Mind, Digital, Proceedings of the Third International Mathematics and Design Conference, Deakin University, Geelong, pp. 56–62.

Chien, S. and Flemming, U.: 1996, Design Space Navigation: An Annotated Bibliography. Technical Report No EDRC 48-37-96, Engineering Design Research Center, Carnegie-Mellon University, Pittsburgh.

Chien, S. and Flemming, U.: 1997, Information Navigation in Generative Design Systems, in Y. Liu, J. Tsou, and J. Hou (eds.), CAADRIA 97, Vol. 2. Hsinchu, National Chia Tung University, pp. 355–366.

Cohen, R., Allaby, C., Cumbaa, C., Fitzgerald, M., Ho, K., Hui, B., Latulipe, C., Lu, F., Moussa, N., Pooley, D., Qian, A. and Siddiqi, S.: 1998, What is initiative?, User Modeling and User-Adapted Interaction 8(3-4), pp. 171–214. URL: citeseer.nj.nec.com/cohen98what.html

Datta, S. and Woodbury, R.: 2002, A Graphical Notation for Mixed-Initiative Dialogue with Generative Design Systems in J. S. Gero (ed.), Artificial Intelligence in Design'02, Kluwer Academic Publishers, The Netherlands, pp. 25–40.

Datta, S.: 2004, Unfolding design spaces interactively, PhD Thesis, The University of Adelaide.

Hybs, I. and Gero, J. S.: 1992, An evolutionary process model of design, Design Studies, 13(3), 273–290.

Jacobson, I., Booch, G. and Rumbaugh, J.: 1998, The Unified Modeling Language User Guide, Addison Wesley, Reading.

Leeuwen, J. V. and Wagter, H.: 1998, A features framework for architectural information : Dynamic models of design, in, J. S. Gero and F. Sudweeks (eds.), Artificial Intelligence in Design '98, Kluwer Academic Publishers, pp. 461–480.

Schön, D. A. and Wiggins, G.: 1992, Kinds of seeing and their functions in designing, Design Studies, 13(2), 135–156.

Smithers, T.: 2000, Designing a font to test a theory, in, J. S. Gero (ed.), Artificial Intelligence in Design '00, Kluwer Academic Publishers, pp. 3–22.

Smithers, T.: 2002, Synthesis in Designing, in J. S. Gero (ed.), Artificial Intelligence in Design '02, Kluwer Academic Publishers, pp. 3–24.

Stouffs, R. and Cumming, M.: 2003, Querying design information through the visual manipulation of representational structures, in, M.-L. Chiu, J.-Y. Tsou, T. Kvan, M. Morozumi, and T.-S. Jeng (eds.), Digital Design: Research and Practice, Proceedings of CAAD Futures'03, Kluwer Academic Publishers, pp. 41–53.

Stouffs, R. and Krishnamurti, R.: 2002, Representational Flexibility for Design, in, J. S. Gero (ed.), Artificial Intelligence in Design '02, Kluwer Academic Publishers, pp. 105–128.

Summers, J., Lacroix, Z., and Shah, J.: 2002, Case-based design facilitated by the design exemplar, in, J. S. Gero (ed.), Artificial Intelligence in Design '02, Kluwer Academic Publishers, pp. 453–476.

Woodbury, R., Burrow, A., Datta, S., and Chang, T.: 1999, Typed Feature Structures and Design Space Exploration, Artificial Intelligence in Design, Engineering and Manufacturing, Special Issue on Generative Design Systems, 13(4), pp. 287–302.

Woodbury, R. and Chang, T.: 1995, Massing and Enclosure Design with SEED-Config, ASCE Journal of Architectural Engineering, 1(4), pp. 170–178.