Supporting designer intention in sketching activities

Conference or Workshop Item

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 99 MUNICH, AUGUST 24-26, 1999

SUPPORTING DESIGNER INTENTION IN SKETCHING ACTIVITIES

David A. Stevenson, Alex H.B. Duffy, Sungwoo Lim

Keywords: sketching, vague geometric modelling, meaning in drawings

1 Introduction

The GEoMetric CONfiguration (GEMCON) project, conducted in the CAD Centre, University of Strathclyde, investigated means of providing appropriate computational support for early design through

- development of a uniform scheme for representing vaguely defined geometric concepts or geometric configurations at multiple levels of detail, suitable for re-use in downstream computer-based design processes, and of a constraint-based reasoning approach to supporting development of geometric configuration solutions [Guan et al, 1997];
- development of a sketcher-based user interface as a platform for investigating appropriate interaction mechanisms for rapid input of vague or concrete geometric information for establishing geometric configuration models, and for editing and presenting such models in ways suited to functional evaluation from multiple viewpoints [Stevenson et al, 1996].

The focus of this paper is the development of the sketcher-based user interface.

2 User Requirements arising from the Modeller

The underlying modelling system developed in the project sought to support a user in establishing geometric configurations using vague, along with concrete, geometric information (including shape, size, location, orientation, and ``nesting'' of geometries), and in gradually evolving the configurations into concrete and precise models. Major user-accessible features include:

- A set of conventional 3D primitive shape types {cuboid, cylinder, frustum, prism, sphere}, each with a user-editable text field, and with orientation specifiable by the user.
- Shape treated as an editable attribute of an object, not a fixed characteristic.
- Object size parameters (e.g. width, radius) hold a range of possible values captured by an interval and user-definable via size constraints such as approx=, <=, range [25,29], etc.
- An object in a geometric configuration is located in a 3D cuboid uncertain region. Until and unless the user chooses to locate that object either by location or relative to some other object (e.g. above, below, right, left, behind, front), the object's position is undefined.
- Objects can contain or be contained by other objects.

3 User Interface Principles and Approach

The primary task of the user interface was seen to be to support the above functionality with a graphical environment facilitating (a) rapid input and manipulation of the above in establishing geometric configuration models, and (b) presentation of the established vague or concrete models. The user interface approach built upon a simple experimental ``pseudo-3D isometric-view sketcher'' ([Aburto, 1993]), offering basic mouse-driven facilities for creating ``layered'' 2D graphical objects presented as isometric views of 3D objects (see Fig. 2).

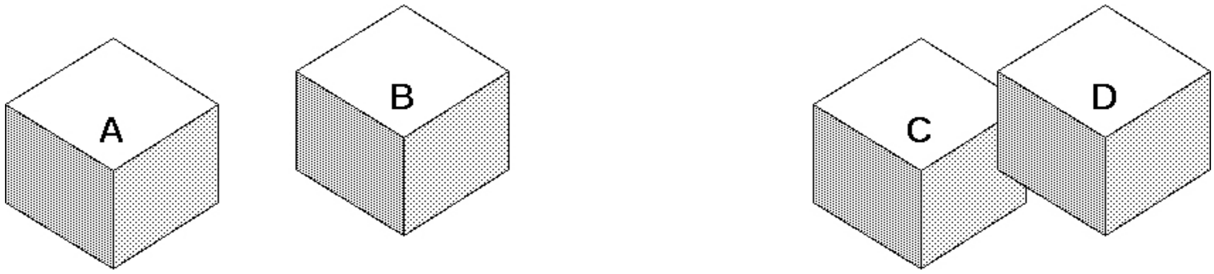

Figure 2. Two separate iso-objects (A and B) and two overlapping iso-objects (C and D)

Two aspects of this approach were seen as particularly relevant to our work:

- economy of computation: pseudo-3D objects would be simple, inexpensive and quick to construct and manipulate (and destroy) - the iso-sketcher could be relatively lightweight
- implicit support for ``minimum-commitment'': there are two things we do not know about objects A and B in Fig. 2: we do not know where either of them is in absolute terms, and we do not know where one object is relative to the other. These ``unknowns'' were seen as implicitly reflecting a situation common in early stages of design: where one or more items of geometry have been created as being, at least provisionally, parts of the proposed design, but with no commitment as to absolute or relative positioning.

The envisaged functionality of the geometric modeller, together with the ``iso-sketcher'' approach, offered a vision of the target interface as a ``sketching'' utility not only supporting underlying modelling of size and location approximation, spatial relationships, abstraction, refinement and shape-changing, but also suggesting support for other iso-3D entities such as ``ribbons'', ``wires'' and ``virtual'' objects (objects whose purpose is to ``reserve space''), and for symbols and annotations such as arrows and free text - entities not necessarily intended to be part of the model itself, but useful to the designer (Fig. 3). Note, however, that not all of this functionality has been realised in the current version of the iso-sketcher.

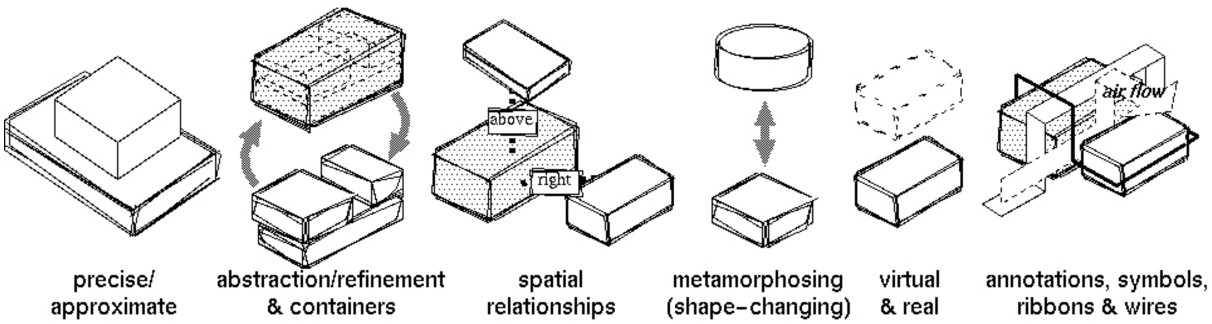

Figure 3. Envisaged elements of the iso-sketcher

4 Development of the Iso-sketcher

The iso-sketcher was built ``from scratch'', using the same Lisp/CLOS environment [Harlequin, 1994] used to build the geometric modeller.

As a first step, a conventional 2D sketcher was built, providing the basic functionality desired for the iso-sketcher, but applying it initially to simple 2D objects only. The 2D sketcher provides conventional flexible mouse-and-keystroke based functionality for creating, resizing, moving, selecting ``raising''/``lowering'' and deleting simple rectangular objects. Each object possesses an editable text label drawn at its centre, and selected objects are attribute-editable in the normal manner using attribute menus; attributes catered for are: fill YES/NO, edge YES/NO, fill colour (if applicable), edge colour (if applicable) and edge thickness. All operations can be invoked by the use of the mouse and menus, and keystroke ``accelerators'' are also provided for some of these operations. All user interaction operations and sequences have embedded ``dummy'' checking routines, for calling out to a ``linked'' application to check the validity of the proposed operation.

The next step was to introduce pseudo-3D objects into the 2D sketcher environment as simply another type of 2D object, subject to all the same constraints and manipulable in exactly the same way as ``conventional'' objects. In particular this required definition of an ``iso-cuboid'' type as the basic iso-object type and methods for creation, manipulation and presentation conforming to those available in the 2D sketcher, but specialised on pseudo-3D objects. Fig. 4 shows the chosen iso-orientation, and examples of the pseudo-3D presentation and functionality. The iso-orientation conforms to a default adopted by the underlying geometric modelling work.

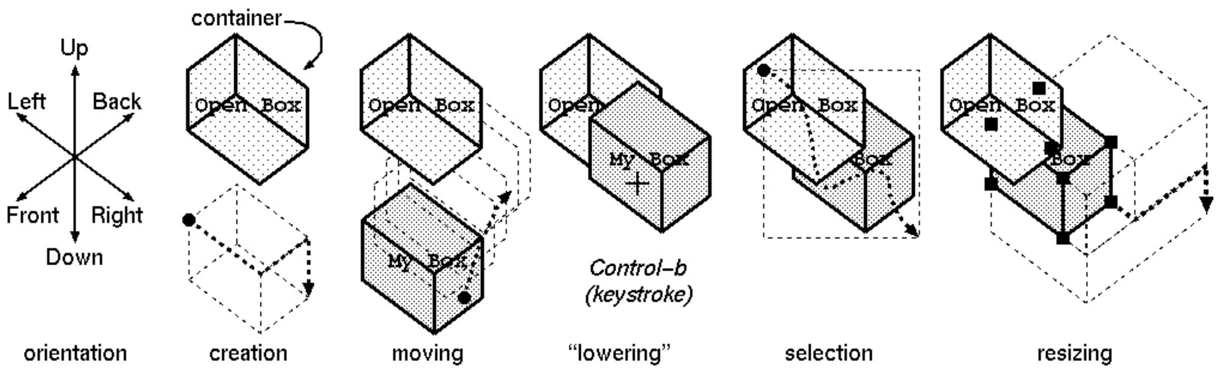

Figure 4. Iso-orientation, containers, and example functionality of the 3D sketcher

4.1 mapping between iso-sketcher and modeller:

Basic mappings were established between iso-objects and geometry objects in the modeller:

- geometries map to editable pseudo-3D iso-objects;
- location constraints and spatial relationships map to simple constraints on the movement and ``raising''/``lowering'' of iso-objects;
- size constraint types map to editable line-type attributes of iso-objects;
- shape-changing and ``containers'' map as editable attributes of pseudo-3D iso-objects;
- abstraction maps to operations on ``groups'' of pseudo-3D iso-objects.

4.2 linking the iso-sketcher to the vague geometry modeller:

The iso-sketcher supports creating, manipulating, editing and deleting of pseudo-3D graphical ``iso-objects". These iso-objects may or may not be ``linked" to underlying geometric entities being processed by the geometric modeller. Where iso-objects are so linked, any operation on them is seen as being, in intent, an operation on the underlying geometric entities, and is processed as a direct call to the underlying geometric modeller. Where such operations influence the state of the underlying geometries (as, typically, they will), these changes are propagated back to the iso-sketcher from the modeller by MINDER as messages, and the iso-sketcher ``edits" affected iso-objects as appropriate. The iso-sketcher never manipulates the iso-objects directly in these circumstances. This provides a clean linkage between sketcher and underlying application, catering in particular for situations where an operation on one geometric entity may affect other linked geometry entities in ways not anticipated by the user. This model is maintained where an iso-object is not linked to an underlying geometric entity, by calling a ``dummy'' checking routine which simply feeds appropriate information back to the iso-sketcher as if sent by the modeller. The ``physical" links between the modeller and the iso-sketcher should be maintained by the MINDER protocol and mechanisms. This part of the work is only partially complete at this time.

5 Discussion - supporting choice in applying meaning to drawings

The approach to the design of the vague geometric modeller and interface sprang directly from the observation that early design is characterised by what is not yet known or decided, and this is reflected in drawing and sketching activity itself. Drawings have no intrinsic meaning - we give them meaning (or not): we map them to mental models as a means of thinking about or understanding a problem. Often we change our mental model to something else, and so have the drawings mean something else. The drawings themselves may not change at all (though it is likely that they will), but what we have them mean has changed. It is observably not uncommon for us to have the same drawings ``mean" more than one thing at a time.

The iso-sketcher's articulated structure and ``linking" mechanism offers support for this way of working: the user can be given the option either to make (or break) such links or not at any time, simply by invoking an appropriate command. In particular, there is no requirement that a graphical ``iso-object" have anything to do with any emerging design model that may be the user's goal in using the sketcher in the first place. The user may ``doodle", in fact. But the user also has the freedom to choose, at any time, to give the sketches meaning: to have them embedded in a context that imposes graphical, mathematical or logical constraints upon their behaviour - in this case, by `linking'' attributes of an iso-object to the underlying modeller.

Nothing intrinsically prohibits a ``sketch" from being linked to more than one computational module at the same time; and this again reflects the way in which we seem to process pencil-and-paper sketches and drawings: we have them ``mean" a number of things at the same time, we use the same drawing to give us multiple simultaneous views onto the problem under consideration. Related to this is the observation that these ``sketched iso-objects" may not even be intended to represent real geometric entities at all: the user might, for example, construct a 3D histogram, and link it to, for example, a dynamic statistical package. The links manifest the interpretations, and carry the meanings which are in force at any given time.

6 Further Work

The major outstanding element of the iso-sketcher work is the setting up the ``linking’’ mechanisms between the underlying modeller and the iso-sketcher. These mechanisms already exist, having been utilised in previous projects, and there is no research involved here. The work has, however, identified various issues worthy of further investigation, including

- user choice in applying and modifying meaning to drawing elements
- multiple views and interpretations
- related concerns such as cognition and interaction
- alternative means of capturing design intent and decisions through `sketching'

This last area in particular is now being followed up in work under the title I-MAGI: Incremental Modelling of Ambiguous Geometric Ideas. The focus here is again on issues of capturing often-vague early design concepts, ideas and intentions in a computational model, drawing both on the Vague Geometry and ISO-sketcher work from GEMCON project and on other recent work in this expanding area (see References). Much of that work is exploring issues of sketch recognition, either from `pre-processed' drawings or from real-time `sketching' activity via various input/output techniques, tools and software; all of it is concerned with issues around the use of sketching to further design activity, and appropriate means of supporting it. The I-MAGI research is examining such work, together with the outcome vague geometry and iso-sketcher work from GEMCON, to further the effort towards powerful, flexible, useful and usable support for sketch-driven design activity.

7 Conclusion

This paper has presented an iso-sketcher, linked to an underlying vague geometric modeller, as a candidate tool to support ``rapid capture and use of geometric concepts in early design''. The key element of the approach has been the use of ``linking'' to make crucial distinctions between the substance of a sketch or drawing (and physical graphical operations applied to it) and any meaning that might be applied to the elements of the drawing and to these operations. The results from the work thus far seem to suggest that in order to provide appropriate, useful and powerful computer-based support for the human activity of design (or problem solving - or indeed thinking itself) using ``sketching'' techniques, it is vital that this distinction be first made and then actively supported..

References

[1] Aburto, L., ``Investigation into Interfaces for Computer-Aided Sketching", CADC/D/93-11, M.Sc. Dissertation, CAD Centre, University of Strathclyde, Sept. 1993.

[2] Branco, V, Costa, A and Ferreira, F N, `` Sketching 3D models with 2D interaction devices", EUROGRAPHICS '94. 1994

[3] Cheng, William Chia-Wei, Tgif - an Xlib based interactive 2-D drawing facility under X11, Copyright (C) 1990-1995, William Chia-Wei Cheng (william@cs.UCLA.edu)

[4] Deering, M F.``HoloSketch : a virtual reality sketching/animation tool", ACM Transactions on Computer-Human Interaction, vol. 2, no. 3, p. 220-238, 1995

[5] Eggli, L, Bruderlin, B D and Elber, G. Sketching as a solid modeling tool. In (eds.), Symposium on solid modeling and applications - Procs. 1995

[6] Gross, M D. ``The Electronic cocktail napkin - a computational environment for working with design diagrams, Design Studies., vol. 17, no. 1, p. 53-69, 1996

[7] Guan, X, Duffy, A H B and MacCallum, K J. ``Prototype system for supporting the incremental modelling of vague geometric configurations", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol. 11, no. , p. 287-310, 1997

[8] Harlequin, The Harlequin Group Limited, LispWorks User's Guide, 1994.

[9] Jenkins, D L and Martin, R. R. ``The importance of free-hand sketching in conceptual design: automatic sketch input", American society of mechanical division, 1993., vol. 53, p. 115-128,

[10] Lamb, D and Bandopadhay, A, ``Interpreting a 3D object from a rough 2D line drawing", In (eds.), Visualisation '90 Proceedings. 1990 Landay, J A and Myers, B A, ``Interactive sketching for the early stages of user interface design, ``In (eds.), Human factors in computing systems (CHI), 1995.

[11] Lipson, H and Shpitalni, M. A, ``new interface for conceptual design based on object reconstruction from a single freehand sketch, CIRP annuals - manufacturing technology, vol. 44, no. 1, p. 133-136 1995

[12] Pugh, D, ``Designing Solid Objects Using Interactive Sketch Interpretation", In (eds.), Computer Graphics (1992 symposium on interactive ed graphics). 1992

[13] Stevenson, D.A., ``An Output-oriented Approach to User Interface Design", M.Phil. Thesis, CAD Centre, University of Strathclyde, 1990.

[14] Stevenson, D A, Guan, X, MacCallum, K J and Duffy, A H B, ``Sketching on the back of the computational envelope ... and then posting it?", Workshop on Visual Representation, Reasoning and Interaction in Design, 22nd June 1996, Fourth International Conference on Artificial Intelligence in Design 1996 (AID'96), Stanford University, USA

[15] Sturgill, M M, Cohen, E and Riesenfeld, R F, ``Feature based 3D sketching for early stage design", In (eds.), proceedings of the Computers in Engineering Conference and the Engineering Database Symposium, ASME. 1995

[16] Zeleznik, R C, Herndon, K P and Hughes, J F, ``SKETCH: An interface for Sketching 3D Scenes", In (eds.), Proceedings of SIGGAPH '96. 1996

David A Stevenson

University of Strathclyde
CAD Centre, DMEM
James Weir Building, 75 Montrose Street, Glasgow G1 IXJ
Scotland
Phone: +44 (0)141-548-3134
Fax:+41 (0)141-552-3148
E-mail: david@cad.strath.ac.uk