University of Warwick
Department of Engineering

Application of Definitive Scripts to Computer Aided Conceptual Design

Alan John Cartwright
MSc CEng MIMechE

A thesis submitted in compliance with the regulations for the award of the degree of Doctor of Philosophy at The University of Warwick. This work was carried out in the Department of Engineering under the supervision of Professor DJ Whitehouse

April 1994

Application of Definitive Scripts to Computer Aided Conceptual Design

Summary

The creative phases of design are based upon the human ability to conceptualise or abstract ideas from physical observations of the real world. That ability comes from experience, based on experiment: discerning patterns of behaviour in particular sets of observations. In this work it is shown that the process of identification, experiment and abstraction may be modelled accurately on a computer by definitive, or agent-oriented, programming, so forming a powerful aid to conceptual design

A new computer modelling language, called EdenLisp, has been developed by the author around Definitive Notations and interfaced to a commercial Computer Aided Design package. It provides a tool whereby computer models of systems can be originated that have state and on which state change can be made, not only by the designer but also by other autonomous agents of change.

Experiments with the language are described that show that scripts of definitions can have characteristics that permit the design to proceed as if there were an engineering prototype of the physical system being designed. The explicit representation of state at the lowest levels permits experimentation, observation of properties and addition of further observations.

The interactive construction of EdenLisp is analogous to the conceptual design process. It is used to illustrate and test design meta-theories for modelling conceptual design. It is shown to have potential for concurrent or multi-agent design, and is also an excellent vehicle for design education.

Contents

Acknowledgements

Interaction, intelligence and integration are siren words for Intelligent CAD systems. They also represent an accurate description of the way that the research group led by Meurig Beynon has helped to develop creative thinking and thoughtful insights that have contributed to this work. I owe an incalculable debt to Meurig and gladly acknowledge all his patience and support over six long years.

Supervision of prima donna academic staff is somewhat difficult, especially with an arcane subject, so I have appreciated Professor David Whitehouse's gentle encouragement and helpful guidance.

Any work of this magnitude takes a slice out of one's life, so I am most grateful for the support and encouragement of my wife Christine, and children Simon and Rebecca who have had their family life dominated by "definitive notations". I am thankful to God, the Great Designer, for the beauty of the world that is His design, and for His care and concern for His children.

Glossary

* indicates a term that is also in this glossary

Abstract Definitive Machine (ADM) A method of structuring a definitive program* into different scripts* each of which can be under the control of an agent; (agents may be acting concurrently). The ADM divides definitive statements into three

- those that the agent can unconditionally redefine*,
- those that the agent can redefine conditionally,
- those on which the agent imposes conditions on the ability of other agents to redefine.

CAD and **CAD/CAM** Computer Aided Design / Computer Aided Manufacture. *Design* is often a misnomer in commercial CAD systems that are actually *Draughting* systems. CAM usually implies a system for producing Numerical Control (**NC**) data to drive machine tools.

CADNO A Definitive Notation for graphics, like DoNaLD* but extended to 3-D.

Chunk A unit of integrated knowledge in long term memory. See also Percept*

Complex, Frame Terms used in this thesis for collections of labels arranged in ways that resemble topological nets and graphs. Nodes and the edges connecting those nodes are associated respectively with labels and the ways that the labels are grouped by means of parentheses.

Computational State In this work the s*tate of a computation* is what may be observed if a computation is suspended at any moment. The significant observations relate to the intention of the program that is running. A static state or statelessness refers to the condition where no observations can be made that were not preconceived by the programmer.

Connectivity The ability of labels to be connected by means of "edges" like nodes in a topological graph. Edges are not geometrically defined; they simply indicate connection. See also complex*.

Content The content of a symbol is whatever may be suggested by that symbol by any person looking at it. Content is always greater than that for which a symbol is used.

Declarative Programming A style of programming where statements attempt to express the desired outcome of the program in terms of "what is" knowledge rather than the "how to" knowledge of procedural programming*. Example are functional languages (Miranda, Lisp) and relational languages (Prolog).

Declarative Knowledge Knowledge about facts.

Definitive Programming (Definitive Notation) A method of programming where program statements take the form of definitions that are to be interpreted in their entirety without regard to their order. When a program, otherwise called a script* of definitions, is input to an Evaluator all the definitions are scanned and evaluated in the manner of a spreadsheet. Addition of further definitions, or redefinition of existing definition immediately trigger the evaluator; the whole script is scanned and re-evaluated as necessary.

Definitions take the form `variable = statement`
where `variables` are conventional computer variables and
`statements` may be values, formulae or functions, or calls to procedures.

DoNaLD is the Definitive Notation for Line Drawing, a notation that permits line based graphics. It translates a script* into EDEN* notation that must then be evaluated by EDEN. CADNO* and SCOUT* are other notations that translate into EDEN.

DXF, IGES, STEP, XBF Intermediate (neutral) codes used to transfer graphical data between different systems, especially between CAD* systems.

EDEN The Evaluator for DEfinitive Notations. A script* submitted to EDEN is evaluated in the form of a generalised spreadsheet.

EdenLisp The definitive evaluator written in AutoLisp, a superset of Lisp, and interfaced with AutoCAD.

Form A symbol used in mathematics or computer programming has an intended meaning in the context where it is used. This is the form of the symbol. Other meanings may be ascribed to the same symbol that were unintended and not in the scope of the application. The latter is the content*.

Frame See complex* or percept*.

GKS, PHIGS Programming software libraries for producing and displaying graphics.

Heuristic A rule for problem solving that is based on the semantic characteristics of the problem rather than its abstract characteristics. It is essentially a search based rather than an algorithmic rule.

ICAD Intelligent Computer Aided Design.

III-CAD The name used by the Dutch CAD research group for Intelligent, Interactive, Integrated CAD systems.

Instantiations are particular instances or examples of an abstract form that are worked out to a more concrete or observable form.

Latent states A definitive script* exists in a particular state (the state of the dialogue*) after it has been evaluated. A change in any definition will change that state. The set of possible redefinitions is infinite but if related to the physical interpretation that set represents all possible or *latent* states of the model represented by the script.

Meta-theory The prefix meta implies a generalisation. A meta-theory is a theory about theories: principles that apply to all theories.

Monotonic reasoning means progressing incrementally from one step to the next by reference to the starting conditions only.

Non-monotonic reasoning is making only one logical step at a time, reflecting on the outcome on the basis of real world observation before proceeding. The result may be a revision of the starting conditions at each step.

OOP = Object Oriented Programming. A style of programming in which programming objects are created that have local state because they have **encapsulated** or hidden variables. Neither those variables nor their values are directly accessible by other objects or procedures. At its simplest such an object may consist of a single procedure. Inputs and outputs connected with the object are of a specific kind allowing standardised linking of objects. An example language is Smalltalk.

Percepts are consistent and lasting ways of grouping observations made by humans on the basis of behaviours that link those observations in ways that seem to be predictable. In the Artificial Intelligence world, percepts are sometimes called **frames.** Frames* are also used in a different way in this thesis.

Procedural Programming A method of computer programming by means of a recipe or set of instructions for obtaining an output from given inputs. Examples are Fortran, Pascal, C.

Procedural Knowledge Knowledge about procedures.

Prototype A prototype in an engineering context is a physical object that is manufactured as the first of the *complete* product of the design process. It is used to check the performance of the product against the specification.

In computer science a prototype is a computer model that is used to check *particular* aspects of the software.

In this work it is used more in the engineering sense.

Redefinition If a definition within a definitive script is revised and resubmitted to the Definitive Evaluator then the whole script is re-evaluated. Redefinition may mean changing its value or totally re-writing it. Provided the redefinition is consistent with the declared types of the variables used any alteration is legal.

Rule-based Systems are often called Expert Systems. Input is referred to an *inference engine* a computer program that causes the input to be tested against the knowledge and rules in the data or knowledge base. The output is a set of hypotheses (preferably of size one) representing a "desirable" outcome.

SCOUT A definitive notation* for interfacing with the Graphical system under X-windows.

Script, (Definitive Script) A set of definitive* statements, roughly corresponding to a conventional computer program.

State "An object is said to have state if its behaviour is influenced by its history. We can characterise an object' s state by state variables, which among them maintain enough information about history to determine the object' s current behaviour." [Abelson *et al,* 1985].

State of the dialogue When using a definitive evaluator the script* is scanned and evaluated after each definition is entered. The display and/or the internal representation reflects the state reached after the last definition was entered. This is called the state of the dialogue.

SDRC I-DEAS A commercial CAD/CAM system for mechanical design - a suite of programs based around a geometrical solid modeller.

TIFF A bitmap coding for transferring graphics data between graphics programs.

Virtual prototype A term used to describe the analogy between a physical prototype* and the computational model of the intended engineering design, that model consisting of a set of definitive scripts*, related according to the rules of the ADM*.