ARTIFICIAL INTELLIGENCE IN ENGINEERING: ROBOTICS AND PROCESSES

EDITOR: J.S. GERO

ELSEVIER
COMPUTATIONAL MECHANICS PUBLICATIONS

ARTIFICIAL INTELLIGENCE IN ENGINEERING: ROBOTICS AND PROCESSES

Edited by

J.S. Gero

University of Sydney, Australia

ELSEVIER
Amsterdam - Oxford - New York - Tokyo 1988

Co-published with

COMPUTATIONAL MECHANICS PUBLICATIONS
Southampton - Boston 1988

J.S. GERO
Department of Architectural Science
University of Sydney
NSW 2006
Australia

Associate Editor: A.D. Adey
Computational Mechanics Publications
Ashurst Lodge
Ashurst
Southampton
SO4 2AA, U.K.

Distribution of this book is being handled by:

ELSEVIER SCIENCE PUBLISHERS B.V.
Sara Burgerhartstraat 25, P.O. Box 211
1000 AE Amsterdam, The Netherlands

Distributors for United States and Canada:

ELSEVIER SCIENCE PUBLISHING COMPANY INC.
52 Vanderbilt Avenue
New York, N.Y. 10017, U.S.A.

British Library Cataloguing in Publication Data
Artificial intelligence in engineering:
robotics and processes.
1. Engineering. Applications of artificial intelligence
I. Gero, John S. (John Steven),
620'0028'563
ISBN 1-85312-011-1

Library of Congress Catalog Card number 88-71240

ISBN 0-444-70470-1	Elsevier Science Publishers B.V.
ISBN 0-444-70466-3(Set)	
ISBN 1-85312-011-1	Computational Mechanics Publications UK
ISBN 0-931215-98-6	Computational Mechanics Publications USA

Published by:

COMPUTATIONAL MECHANCIS PUBLICATIONS
Ashurst Lodge, Ashurst
Southampton, SO4 2AA, U.K.

Printed in Great Britain by The Bath Press Limited, Avon

ELSEVIER

COMPUTATIONAL
MECHANICS
PUBLICATIONS

Acknowledgement is made to M.R. Jahanbin and F. Fallside for the use of Figure 1a) on page 124, which appears on the front cover of this book.

PREFACE

The International Conference on Applications of Artificial Intelligence in Engineering series is establishing itself as the unique forum for the presentation of the latest research, development and application of artificial intelligence in engineering. It covers all the engineering fields. The first conference was held in Southampton, UK in 1986. The second conference was held in Cambridge, Massachusetts, USA in 1987. The third conference was held in Palo Alto, California, USA in August 1988. This volume and its two companion volumes contain the papers presented at the third conference.

Computers and computing have become an integral part of engineering. Computers are used extensively to carry out analyses and occasionally in low level design such as member selection in structural engineering. They are being increasingly used to produce drawings and other graphical representations to communicate ideas. Most engineering offices also use computers for office management, job control and word processing. All of these applications are built on two fundamental concepts. The first concept is that the world of interest can be described by the "calculus of real numbers—algebra". The second concept has to do with computing itself and is concerned with the way we make computers work, through instructions to the computer codified as "procedures" which describe what must be done. These we call computer programs—a set of detailed instructions which are executed sequentially unless the program contains instructions which alter this sequential flow. Whilst this is a concept deeply embedded in traditional computing it has important ramifications for users. Such computing is called von Neumann computing after the inventor of this architecture for computing.

Engineering is concerned with much more than calculation based on mathematical descriptions of the world. Professional engineering is concerned with concepts, ideas, judgment and experience. All of these appear to be outside the realm of traditional computing. Human beings discourse with each other using models of the worlds largely unrelated to either mathematical descriptions or procedural representations. They make use of knowledge about objects, events and processes and make declarative statements about them. These are often written down symbolically. The limits of traditional computing are that it is unable to represent and manipulate knowledge in an explicit and coherent form and that it is unable readily to perform symbolic computation.

Artificial intelligence is largely concerned with the acquisition, representation and manipulation of human knowledge in symbolic form. Human knowledge is thought of as being reasoning (rather than the simple ability to acquire facts as you might find in an encyclopedia). Just as the industrial revolution can be considered to have automated mechanical power, and the computer revolution to have automated calculation, so artificial intelligence automates symbolic reasoning.

Barr and Feigenbaum in *The Handbook of Artificial Intelligence* could well have been writing about professional engineering knowledge when they stated: "Since there are no mathematical cores to structure the calculational use of the computer, such areas will inevitably be served by symbolic models and symbolic inference (reasoning) techniques."

Robotics has the potential to change the physical face of engineering and has no counterpart in traditional engineering. It requires the integration of numerous disparate aspects of engineering. There is a fundamental requirement for geometric and spatial reasoning of a qualitative kind. A variety of other processes in engineering are being examined through the artificial intelligence lens. The material collected under the process rubric demonstrates both the benefits and potential of utilising this approach.

The papers in this volume are presented under the following headings.

Robotics
Geometric and Spatial Reasoning
Interpretation Processes
Reasoning Processes
Planning and Scheduling Processes
Interfaces

A large number of papers were submitted for consideration. Each paper was refereed by three referees. This task was ably executed with the aid of the International Advisory Board listed below. Members of the board worked assiduously to select appropriate papers, thanks are due to them.

The final manuscript was sub-edited by Fay Sudweeks to produce a degree of uniformity lacking in the submissions.

John S. Gero
University of Sydney

CONTENTS

SECTION ONE

Robotics

Functional reasoning for flexible robots

D. Tezza, E. Trucco
DIST, University of Genova, via All'Opera Pia 11A, 16145 Genova, Italy

ABSTRACT How can a robot use tools intelligently? Why do we use different objects to perform the same function? Effective answers to such questions could provide a dramatic improvement for intelligent manipulators and are based on the notion of *functional reasoning* as the ability of integrating shape, function and planning in reasoning about the world. This paper is an insight into the problem. It outlines the FUR project, an attempt to develop a computational model for functional reasoning, apparently ubiquitous in human behavior. Relevant computational literature is reported and discussed. The development state of the project is presented along with the implementation of a first prototype. Some experimental results are finally given.

INTRODUCTION: THE CONCEPT OF FUNCTIONAL REASONING

In everyday life, we show a puzzling ability in making use of the same object in many different ways. This suggests that a mental model is employed which associates that shape to several functionalities. As Freeman and Newell pointed out in a pioneering work (Freeman and Newell [1]), it is not difficult to notice that "a pencil can print characters, punch holes in paper, serve as core for a ball of string and tamp down pipe tobacco". The shape of the pencil suggests us the actions mentioned above; conversely, once decided to punch a hole in a paper sheet we look for a pencil and not for, say, a rubber. In this paper we introduce FUR (Di Manzo et al. [2, 3]), a computational model integrating structural and functional information which accounts for such behaviors. This kind of representation is responsible for the generation and refinement of a number of common actions involving tools, that's objects designed to accomplish a well−defined function, and other man−made objects. As Brady and Agre point out (Brady et al. [4]), tools representations participate in plans and evolve with them, adapting to new situations, fixing the bugs of old tools and generating new objects and plans. To define exactly what a tool is, is an unexpectedly puzzling problem. A wide analysis from the point of view of biology is given in the fascinating book by B. Beck [5].

Many examples could be given which show the ubiquity of functional reasoning, e.g. tool design ("the art of devising artifacts to attain goals", as Simon [6] characterized it), or simple activities as choosing the right tool for a job, or again classifying objects according to the function that their structure suggests. We often name things after the functions they provide: a "washing machine", a "screwdriver" and so forth.

A proper model for the shape—function relation could provide powerful aid to an intelligent robot, making vision and planning processes more flexible. Functional information can be associated to geometric object models, thus allowing more general object classification and recognition. On the other hand, knowing what a tool is for could enable manipulators to infer how to use even different tools to achieve the same goal. This points are discussed in the rest of the paper. In order to provide a useful degree of generality, a knowledge representation model should be far more complex than a simple one—to—one relation between actions and objects, e.g. hit—hammer, sit—chair and so on. Moreover, functional descriptors of objects introduce an explicit link between structures and actions and therefore with the plans involving the represented objects (Brady et al. [4,7]). In other words, the problem is how to represent the relation between the structural properties of objects and their function, which is in turn related to the actions they make possible.

RELATED WORK

In the seminal work by Newell and Freeman [1] objects were classified according to their function. In their model, composing parts to get complex structures was possible if a part performed the function that another part required. Matched functions were "consumed" inside the composed structure and not available for the resulting structure any longer. Two strategies were indicated for parts composing: forward chaining and backwards chaining. The former considers how to build a structure fulfilling a required function starting from a set of given objects; the latter how to build an assembly starting from the required function down to match the functional properties of single objects. Indeed this work shed light into the field of automated functional reasoning, but did not identify a general model for the shape—function relationship as well as a well—defined role for visual data.

Winston et al. [8] have suggested that functional descriptors could be learnt in terms of structural features. They discuss the following advantage of the functional approach: it is often too hard to tell vision systems what things look like; it can be easier to talk about purpose and link purpose to functional constraints. This means that an unique descriptor could identify an object class and be used to recognize instances which look even very different from one another, provided they are recognized to perform the same function. A set of functional constraints was therefore added to structural descriptors. The system they developed used ACRONYM to drive basic structural descriptions of 3—D objects and matched them to a functional model to drive recognition and learning.

Ingrand and Latombe [9] have developed a system, SERF (Expert System for Functional Reasoning), where complex structures are deduced from functional constraints, that is a set of properties related to the object function. Assemblies are generated composing simpler structures and functional constraints propagated through the resulting levels. Here too a real paradigm for the shape—function relationship seems to be missed. The basic elements considered are rather complex mechanical parts whose shape and function are taken for granted. Each basic element is associated to a number of functions providing a vocabulary for functional reasoning.

Brady (Brady et al. [4]) introduces the "Mechanic's Mate" project. Its first aim is to understand the interplay between planning and reasoning that involves tools and fasteners. This comprehends how function can be deduced from shape and vice versa, what Lowry called "reasoning between structure and function" (Lowry [10]). An advantage of this kind of description has been pointed out in terms of "functional improvisation" (Brady et al. [4, 11]): e.g., deducing that the handle of a screwdriver can work as hammer head if needed. Another interesting point is that the shape–function relation is regarded as closely dependent on the dynamic representation of the world, which can be given in terms of naive physics models (Forbus [12, 13], deKleer [14]). This point will be considered again, discussing the restrictions of the FUR representation. The basic tenet is that it is often difficult to separate the function of a tool from the plan in which it participates and with which it coevolved: plans and tools evolve together and differentiate with time.

THE FUR MODEL FOR FUNCTIONAL REASONING

It is interesting to notice that some tasks involving functional reasoning require manipulation of solid shapes, while others can be formulated at a different abstraction level, rather independently of 3–D geometry. E.g, when you feel thirsty it is immediate to formulate a plan whose first steps are probably going to the kitchen and looking for a glass or cup. The formulation of such a plan per se is quite independent of the shape of the particular container that you will actually find in the end, while geometric reasoning has a definite importance in the subtask of identifying a glass in the crockery shelf. At the same time, the functional definition of glass is what makes you involve a glass in your plan.

This suggests a hierarchy of levels for reasoning between shape and function, adopted in FUR and sketched in fig.1.

Geometric reasoning takes place on a 3–D representation of a scene. This is the level of geometry: a vocabulary of *geometric primitives* is defined, which compute structural properties of bodies as e.g. concavities, baricentrum, volume and so forth. They constitute the interface between the 3–D model and the function level, in that primitive functions (see below) are represented solely in terms of entities of this level.

The *function modeling* level describes a set of *primitive functions* in terms of geometric constraints. Primitive functions are implemented by *functional experts* which collect information about the conditions under which a function is possible. As detailed later, these conditions can be expressed through sets of geometric constraints for a number of common functions. In this sense, one typical activity of the FUR system is to decide whether an object can accomplish a given (maybe primitive) function, e.g. whether a bolthead can be grasped by a given spanner, or to find all graspable objects in a scene.

The *object representation* level uses Functional Experts and geometric relations to describe objects. This is accomplished composing Functional

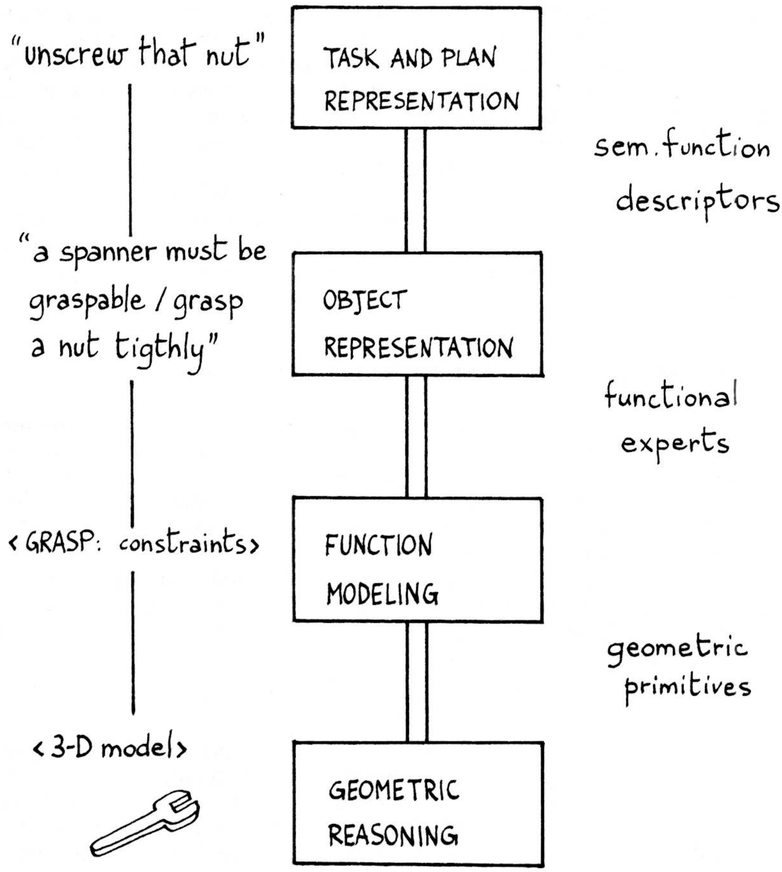

Figure 1. Levels of the FUR model.

Experts to build *semantic functional descriptors* describing objects by semantic networks which integrate functional and structural properties (e.g., a hierarchical structural decomposition along with the function supposed for each subpart).

The *plans representation* level uses Semantic Function Descriptors and Functional Experts for function−driven planning. This comprehends generating or recognizing plans on the functions they involve. At this level, primitive actions are regarded as the basic planning element (say an *elementary plan*) which can be directly connected to geometric constraints through the corresponding Functional Expert. This level has not yet been investigated and will not be discussed in this paper.

The FUR theory is being developed under two main assumptions:

1) the functional goal of most man−made objects can be expressed in terms of some very common, basic functionalities, called *primitive functions*.

2) most man−made objects can be hierarchically decomposed in subparts. Each Semantic Function Descriptor suggests a hierarchical decomposition for the object it describes, associating each subpart to a single basic function. As pointed out before, however, the one−to−one relation between the shape of a subpart and its function is *not* satisfactory. The same shape can accomplish different functions in different objects, that's in different functional contexts. Incidentally, this is what makes functional improvisation possible. FUR accounts for this through Functional Experts. Details are given in the next section.

THE GEOMETRIC REASONING LEVEL

We suppose that a volumetric model of a scene is available, as it could be generated by early processing of a stereo pair or a sequence of images of the same scene (see for instance Herman and Kanade [15], Grimson [16], Morasso and Sandini [17]). Many FUR tasks take this representation as their input data, e.g. primitive function recognition or function−driven object identification. *Octrees* (see e.g. Meagher [18], Jackins and Tanimoto [19], Chen and Huang [20]) have been employed for modeling solid bodies. This choice depends on the fact that octrees are a well−understood, highly structured scheme for 3−D solid modeling; and, most importantly, they seem a suitable way to represent data obtained from early vision processing. Fig.2 shows an example of octree modeling a 3−D scene. An octree−based solid−modeler has been implemented. It enables a user to create 3−D objects and environments from scratch or to manipulate them. For the time being, the scenes considered for analysis do not consider the degradation due to the noise introduced by early vision processing.

The entities in the Geometric Reasoning Level are divided in three groups:

1) *geometric experts*;

2) *geometric functions*;

3) *geometric predicates*.