Proceedings of the Fifteenth ISPE/IEEE International Conference on CAD/CAM,
Robotics and Factories of the Future, CARS & FOF'99
Eds. M F Carvalho and F W Muller, Vol. 2, pp6-11,
Aguas de Lindoia, Brazil, 18-20 August, 1999

AN INTENT-DRIVEN PARADIGM

FOR FEATURE-BASED DESIGN

M.S.Hounsell[1] and K.Case[2]

[1] Computer Science Department, State University of Santa Caterina, UDESC, Joinville, Brazil, 89223-100
[2] Department of Manufacturing Engineering, Loughborough University, Loughborough, UK, LE11 3TU

Email: *dcc2msh@joinville.udesc.br*

Abstract: A very important advantage of a feature-based modelling (**FBM**) system is claimed to be its ability to capture and carry designer's intents (**DI's**), although this last term is rarely clearly defined. Feature's extra non-geometrical semantics, that are closely related to such designer's intents, are used by many applications but never related back to designer's intents. Therefore, adopting the approach of defining of designer's intents helps define the role of features in the geometric design and, indeed, allows future feature-based modelling systems to better represent, store and reuse such information. Moreover, it allows a more formal approach for manipulating, verifying and maintaining DI's throughout the design process, which is an invaluable support for really intelligent CAD systems. This paper presents Designer's Intents in the feature-based modelling context and exposes a methodology used to effectivelly capture and manage and verify this extra information.

Keywords: Feature-based Modelling, CAD, CAM, CAE, Feature-based Reasoning, Designer's Intent.

INTRODUCTION

Conventional **CAD** systems have promoted the marriage of an efficient Geometric Solid Modelling (**GSM**) system, such as Boundary Representation **(B-rep**), Constructive Solid Geometry (**CSG**) or, a hybrid approach (Zeid91), with user-friendly graphical interfaces composing successful systems to help the detailed design activity. Integrating or even interfacing conventional CAD systems with other activities such as engineering (**CAE**), process planning (**CAPP**), manufacturing (**CAM**) and, production & control (**CAPC**) has been shown to be a difficult task because conventional CAD systems are incapable of capturing non-geometric aspects of the designer's intent such as tolerances, part relationships, surface finish, etc. (Nnaji93, Stroud93, Marefat93b). Also, more abstract design activities such as conceptual design, generation of design alternatives, reuse and reasoning on design procedures and, capturing the functionality of a product are just impossible (Henderson93, Taylor96).

Capturing DI's at early stages of the design in a more user-friendly interface that includes a meaningful vocabulary to the designer is a property of a Feature-based Modelling (**FBM**) system that allows more intelligent decisions and reasonings to be made and this is considered "the only possible basis for Intelligent CAD" (**ICAD**) systems (Dixon90).

Future CAD technology is likely to be based on Feature-based Modelling (Mantyla96). Designer's intents represent information that should be verified and maintained throughout the detailed design process and could be used to drive the decision-making for downstream applications. Because they are considered intrinsic to features, they are sometimes omitted from the formal and explicit description of a design. Nevertheless, Feature Based Designer's Intents (**FbDI's**) act as a suitable medium for the validation of feature-based representations.

The main objective of this study was to distinguish and separate the geometrical, factual and intentional feature-based data from its use and interpretation by an application. In doing so this study provides a solution to reduce complex intent-driven engineering reasoning to a more atomic level. This information can then be used as a foundation for a much clearer and powerful reasoning within a Design-by-Feature (**Dbf**) system. This paper presents how this information (FbDI's) has been defined, classified, verified and used to raise the intellingence of a FBM CAD system. In addition, because DI's will be a separate entity it will be easier to store, manipulate and reuse this information by any application. The paper also shows how FbDI's can be explicitly and consciously assigned to the design by the designer (through a direct instantiation or confirmed automatic recognition).

DEFINING DESIGNER'S INTENTS

It has been acknowledged that "the information that constitutes intent, and how to capture and use intent are all research issues to be explored" (Dixon90). Feature-based Designer's Intents (**FbDI's**) was defined as representing a wide variety of concerns that help decide on a specific geometric attribute or configuration. They are factual peculiarities of the geometric design that are intrinsic to features themselves or to the use of features in the design and have engineering-related purposes. FBDI's are properties that are expected to arise in the model because of the use of a feature in a specific location or because of the interaction that a feature provokes with the existing surrounding features in the model.

The exhaustive enumeration of all possible sets of FbDI's is a very cumbersome approach even in a limited domain, and so the objective was to explicitly categorise FbDI's in such a way that this extra information could be effectively and consciously instantiated into a model. In this way the capturing, verifying and maintaining of FbDI's could be performed by, and even automatically discovered by, a design-by-features system. A taxonomy of such intents has been defined and detailed in (Hounsell98). The following sections briefly outline some of the more important aspects.

FEATURE-BASED DESIGNER'S INTENTS (FBDI'S)

Feature-based Designer's Intents are characterised as Theoretical, Relational or Morphological. Each of these types has a set of objectives and a tangible set of properties to enable their implementation within the geometric realm. The generic types specify general engineering concepts or behaviours while the specific FbDI's are computable relationships between features themselves or elements of the feature-based model such as feature faces (and their attributes) or feature parameters.

Theoretical Functional FbDI's

Features may have a *function* concept itself which is defined as "the behaviour of an object, an operation of energy, material, information or signal that tells what the design does" (Tomiyama93) and, "include not only in-use purpose, but also manufacturing and life-cycle considerations' (Dixon90).

The relationships between *form* and *function* cannot be formalised because of many difficulties (Solomons) including the abstract nature and understanding of the *function* concept, the fact that functionality can be a composite result of many interacting sub-functions, and that a given *function* could be performed by several forms and one form might be used to perform a number of different *functions*.

This *function* concept has been implemented as physics-based or engineering-based laws, rules or formulae depending on the underlying theory such as heat propagation, torque or force transference or, stress analysis. Thus, they are called *theoretical functional* FbDI's.

Theoretical functional FbDI's are intents that make specific shape aspects appear on the part's surface, control the part's overall outlook and, are driven by a close relationship between a feature's theoretical functional behaviour and its form. This is possible by manipulating and controlling the hierarchy or dependency of parameters that establish dimensions, profiles (e.g.: quadric, circular, spherical), parameterised local operations (blending, chamfering, trimming), and so on. Theoretical functional FbDI's can be achieved via a parametric constraint-based approach and therefore are not discussed any further.

Relational Functional FbDI's

While theoretical functional intents are usually expressed by formulae, engineering constraints are expressed in the form of relationships between design entities. Thus, they are called *relational functional* FbDI's (RDI's).

Relational FBDI's comprise different disciplines and are dependent on the application of the feature-based model. Relational FBDI's are mostly geometrical facts that have a functional significance for an application. For instance, a "*nested at the bottom*" relational FBDI is a geometrically based and provable fact that could be used by a CAPP system to establish machining precedence among features. RDI's describe physical and/or spatial relationships between features and are categorised as being application-dependent but primarily geometry-dependent, in which case they are called Geometric RDI's (**GDI's**), and geometry-dependent but primarily application-oriented, called Application Oriented RDI's (**AOI's**).

The importance of GDI's has been recognised by many systems that incorporate spatial reasoning in various ways (Silva90, Nelsen91 Vancza93). GDI's are geometrical facts and intentional relationships between entities of a feature-based modelling system but they alone do not suffice for an application.

Positional GDI's include concentric, opposite, planar, coplanar and concentric intents between features. Orientational GDI's include parallel, perpendicular, angularity, against, co-linearity and common External Access Direction intents. Hierarchical GDI's include nested at the bottom and nested at the side. Structural GDI's include patterns with linear, circular, planar or spatial distribution; radial, axial or mirror-like symmetry and co-radius intention.

Application-Oriented RDI's (**AOI's**) arise from the intentions of manufacturing engineers, process planners, etc becoming a part of the design information. Many of these intents are concerns to be fulfilled that guarantee the physical realisation of the design constrained by pragmatic and technological requirements such as cost, quality, time, accessibility and feasibility.

Application-oriented FbDI's include: same or different set-up intents; parent-child and precedence intentional relationships; T-slot, cross feature, entry feature, counter-bore, counter-sink and cut-out compound intentions between features and thin-wall proximity intentions.

Morphological Funciotnal FBDI's

Features represent a good means to embed functional significance into the geometric detailed design phase and this fact can be inferred by some definitions applied to features. Features have been defined as the addition of functionality to geometric forms (Dixon90, Sodhi91, Nnaji93), high-level morphological information with well-defined functional meaning (Gomes91) and, high-level functionally significant entities (Laakko93, Bronsvoort93).

The extra descriptive factors that are added to the topological and geometrical aspects of the geometric solid model are frequently used to better specify the elements of a feature family. Thus Neilson and Dixon (Nielsen91) describe how a *cylindrical boss* family of features could be specialised into a *disk* for a certain height-to-diamenter range or into a *rod* with an alternative ratio.

Hence features clearly have a *morphological function*, which in the geometric domain have been implemented as Volumetric Designer's Intents (**VDI's**) to define expected geometric behaviour FbDI's for features.

Four Volumetric Designer's Intents (**VDI's**) are of particular interest. The *labelling* VDI identifies the relationships between all the feature's faces and their attributes. The feature's additive or subtractive *nature* implies that a change in the feature-based representation must result in a change in the volume and surface of the component being modelled. This requirement and the ability of a feature to change the existing model is called the *changeability* VDI. A feature must have adequate parameters to exactly fit and define the intended form (in the same way as an edge is limited by its two exact ends, called vertices) thus, the feature must fit within the limits of where it is intended to be placed. This ability to fit is called the *fittability* VDI. Furthermore, interesting and difficult situations arise when redundant intents are found. Features that have overlapping volumes usually present a *redundant* VDI.

FRIEND, AN INTENT-DRIVEN SYSTEM

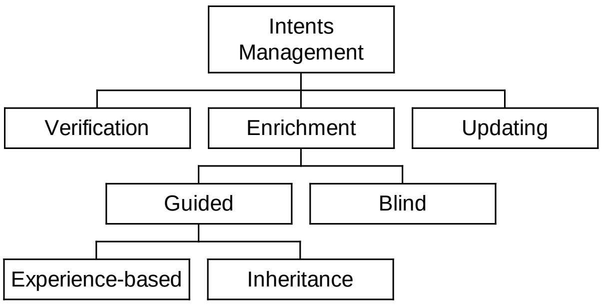
An intent-driven paradigm would suggest that a feature-based modelling application could reason not only <u>with</u> designer's intents (as most of the applications surveyed claim to do, e.g. Zhang93), but could be able to reason <u>about</u> designer's intents. Therefore, means to validate, manipulate and manage FBDI's are required.

A prototype system has been implemented, called **FRIEND** (a **F**eature-based validation **R**easoning system for **I**ntent-driven **EN**gineering **D**esign), using FbDI's as a resource for its reasoning and to perform the conceptual validation of the feature-based representation. To achieve this task **FRIEND** deploy its reasoning following a Intents Management strategy.

Intents Management

Conceptual feature-based representation validation is performed via Morphological Functional (**MFI**) reasoning (details can be found in Hounsell97). MFI reasoning is not only responsible for identifying invalid morphological situations and deploying revalidation operations but also for *adding* and *deleting* VDI relationships. Occasionally, it is possible that RDI relationships may be created by MFI reasonings. This suggests that MFI reasonings drive some RDI reasonings. However, there are RDI reasonings that are independent of MFI and feature interaction cases. In other words, there are RDI's that are dependent on their own functional meanings and therefore have their own reasonings.

Figure 1: A Classification of Intents Management



Rules.

Three ways can be identified to manage FbDI validation (Figure 1): *verification*, *enrichment* and *updating* statements.

Verification Statements

The *verification* statement is used to check if the assigned FbDI in the model complies with its conditions. Otherwise, it can lead to its removal from the model.

The general outline of *verification* rules is depicted in Figure 2 where:

- "!" means existence or true/active;
- "~" means absence or false/inactive;
- "FbDI" is the target feature-based designer's intent;
- "Cond" is a condition being tested.

Verification Rules
IF (FbDI)! AND (~$Cond_1$ OR ~$Cond_2$ OR …)
THEN Ask "DELETE the FbDI?"
  YES, *Delete* FbDI
  NO, Operate Features

Figure 2: An Outline of *Verification* Rules.

Enrichment Statements

The *enrichment* (or feature intent recognition) statement works in the opposite direction to a *verification* in that it analyses a set of conditions and assigns its findings to the model (automatically or assisted by the designer).

Verification statements are basically invalidity tests that inactivate a FbDI as soon as any of its conditions are violated. *Enrichment* statements do the opposite by considering a set of conditions that suggest a FbDI to be assigned to the model. However, two ways can be identified to perform such a search: via *guided* rules or via *blind* rules of *enrichment*.

Blind rules of *enrichment* involve trying a FbDI relationship against all possible situations using a minimal condition set and leaving the confirmation task to the user. This approach is likely to identify an important FbDI but also leads to a tedious confirmation task.

Guided *enrichment* rules search for FbDI's where they are more likely to occur through rules that include basic conditions plus other conditions identified by experience. Although a less tedious confirmation process follows, it is possible that a FbDI can be omitted from the model due to an inaccurate or missing rule.

Guided *enrichment* can be further classified () into *experience-based* guided enrichment when they are isolated rules as mentioned above and *inheritance-based* guided enrichment when the rules are embedded and dependent on other (mainly VDI) reasonings.

For instance, it is sensible to think that features that were split from another tend to inherit the former's FbDI's. The general outline of *enrichment* rules are depicted in Figure 3:

Enrichment Rules
IF ~(FbDI) AND ($Cond_1$ AND $Cond_2$ AND …)!
THEN Ask "ASSIGN the FbDI?"
  YES, *Add* FbDI
  NO, nothing to do

Figure 3: An Outline of *Enrichment* Rules.

Update Statements

Verification and *enrichment* are responsible for *deleting* and *adding* FbDI relationships to the model, respectively.

In addition to *enrichment* and *verification* statements, other rules are necessary to help the management of FbDI's. These are called *updating* rules (Figure 1) and they consider the status of the features involved in the relationship and activate or inactivate FbDI's accordingly. Examples of *updating* rules include: if a feature that previously made obsolete or deleted another feature is subsequently inactivated then the latter should be reactivated and the corresponding VDI inactivated.

Active, Inactive and Intentional Status

The process of design can cause the representation to go through many intermediate stages. One approach to help cope with these intermediate stages is to define an *intentional* or dormant status (which is compatible with the intent-driven terminology).

The idea of intentional features have been already introduced (Tomyiama90): "Intentional features, originally identified by the designer, should not be confused with their geometric embodiments which can vary as the model is edited". This distinction is essential for representing and interrogating invalid features and helps the tracing of feature evolution through the life-cycle of a design model.

Similarly, the intent-driven validation framework (Hounsell96) makes use of the *intentional* status and thus features (or more precisely, their *volumetric intentions*) and FbDI's are kept in the framework in one of three possible status:

- The *active* status, which accounts for all those features and FbDI's that represent the actual model. After the reasoning is finished, the active status identifies validated (non-invalid) features and validated FbDI's.
- The *inactive* status, which refers to all features and FbDI's that were deleted by the reasoning of interacting features or by the user and are not

affecting the actual model. Inactive features and FbDI's explicitly deleted by the user will not become active in the future and can be effectively removed from the database. An *inactive* FbDI means that the possibility of existence of this FbDI was considered before, presented to the user and discarded. In this case, an *inactive* FbDI would have been created to flag the discarded attempt and will not be considered subsequently so long as the conditions do not change.

- The *intentional* status accounts for dormant or intermediate situations. *Intentional* features are those that were made obsolete by another feature. Their *volumetric intentions* still affect the model but are encompassed by the *volumetric intention* of another feature. If this second, volumetrically encompassing, feature is removed the former *intentional* feature should be activated.

DISCUSSION

FbDI's can be used to reason about the design knowledge and structure and are not restricted to the derivation of parameter or dimension values. FbDI's are thus considered a generalisation of constraints where not only fixed algebraic and geometric relationships are considered but also other engineering-related relations (such as *nested@side* GDI and *x_feat* AOI) are included.

Not all FbDI's have both *verification* and *enrichment* statements. *Enrichment* statements in particular are hard to conceive and, although possible in some cases, they are often not practical. For instance, Parametrical FBDI's are composed essentially by *verification* statements.

It is expected that a FBM system driven by FBDI's could give better support to help preserve the reasons for a particular decision in a design. For instance, the reason for a feature to be located at a specific position could be the *axial symmetry* geometrical RDI to be achieved.

It was found that a comparison between the functionalities of the prototype system implementation **FRIEND** and other systems is not straightforward because most of the systems studied perform some variety of geometric reasoning on the *complete* model (and therefore, as a post-processing procedure) while **FRIEND** accumulates knowledge *throughout* the design process by analysing the part model every time an operation is performed. An evaluation of this methodology can be found in (Case99).

CONCLUSION

Feature-based Designer's Intents (**FbDI's**) were divided into three areas: related to individual features (the VDI's); related to groups of features (the geometrical RDI's and PDI's), and; dependent on applications (the application-oriented RDI's).

A better understanding and categorisation of Designer's Intent (**FbDI's**) meaning within a feature-based CAD system is a necessary step to foresee how feature-reasoning could be embedded into future Intelligent CAD systems.

Automatic recognition of pre-established FbDI's and the consequent representation *enrichment* can be achieved, raising the quality and usefulness of the model as well as relieving the designer of these tedious tasks. Feature-based intent recognition, or representation *enrichment*, is a powerful resource that would facilitate "intelligent" reasoning.

The research demonstrates that it is possible for a FBM system to effectively and explicitly represent, capture, manipulate and use designer's intents for reasoning during on-going design. A clear way of defining, identifying and analysing valid and invalid model representations based on FbDI's has also been presented.

The prototype system implemented, **FRIEND** (a **F**eature-based **R**easoning system for **I**ntent-driven **En**gineering **D**esign), is capable of reasoning Morphological Functional FBDI's as well as some Relational Functional FBDI's. **FRIEND** not only verifies the representation but also is capable of suggesting enrichments that could be then maintained and verified. Details of implementation and an evaluation of this validation methodology has been presented in Case99.

REFERENCES

Bronsvoort, W. F. and F. W. Jansen. "*Feature Modelling and Conversion - Key Concepts to Concurrent Engineering*". Computers in Industry, Vol. **21**, pp. 61-86. **1993**.

Case, K and Hounsell, M. S. "*An Evaluation of a Feature Modeling Validation Method*". (to be publised) 15^{th} International Conference on Computer Aided Production Engineering, April, University of Durham, Durham, UK. **1999**.

Dixon, J. R. and J. J. Cunningham. "*Research in designing with features". IFIP WG 5.2, Conf. on Intelligent CAD*, Boston, Vol. **1**, pp. 137-148. **1987**.

Dixon, J. R., E. C. Libardi Jr, and E. H. Nielsen. "*Unresolved Research Issues in Development of Design-with-Features Systems*". IFIP WG 5.2/ NSF Working Conf. on Geometric Modelling, ResselaerVille - USA, Elsevier Sci. Pub., Vol. **1**, pp. 183-196. **1990**.

Dohmen, M. "*Constraint techniques in interactive feature modeling*". TUDelft - Delft University of Technology, Faculty of Technical Mathematics and Informatics, **1994**.

Gomes, A. J. P. and J. C. G. Teixeira. "*Form Feature Modelling in a Hybrid CSG/Brep Scheme*". Computers and Graphics (Pergamon Press), Vol. **15**(2), pp. 217-229. **1991**.

Henderson, M. R. "*Representing Functionality and Design Intent in Product Models*". Second Symposium on Solid Modeling and Applications, May 19-21, Montreal, Canada, Vol. **1**, pp. 387-396. **1993**.

Hounsell, M. S. and Case. "*Representation Validation in Feature-based Modelling: A Framework for Design Correctness Analysis and Assurance*". 12th National Conference on Manufacturing Research Vol. **1** pp. 255-261. **1996**.

Hounsell, M. S. and Case. "*Morphological and Volumetrical Feature-based Designer's Intents*". 13th National Conference on Manufacturing Research Vol. **1.** D. K. Harrison (ed.), Glasgow, Scotland, Vol. **1**, pp. 64-68. September, **1997**.

Hounsell, M. S. and Case. "*A Taxonomy of Feature-based Designer's Intents*" *(IMC'98).* 15th Conference of the Irish Manufacturing Committee, Belfast, Ireland, September, pp. 193-202. **1998.**

Case, K and Hounsell, M S. "Evaluation of a Feature Modelling Validation Method" (CAPE'99). Proceedings of the 15th International Conference on Computer Aided Production Engineering. ISBN: 0 9535558 0 1. Editors: P.G.Maropoulos and J.A.McGeough. University of Durham, School of Engineering, UK, 19-21. **1999**.

Laakko, T. and M. Mantyla. "A feature definition language for bringing solids and features". Second Symposium on Solid Modeling and Applications, May 19-21, Montreal, Canada, pp. 333-342. **1993**.

Marefat, M., *et al*. "*Capturing Intelligence as a Reusable Framework for Manufacturing Decision Processes*". International Journal of Production Research, Vol. **31**(8), pp. 1767-1795. **1993**.

Mäntylä, M., K. Lagus, T. Laakko and G. Sohlenius. *"Application of Constraint Propagation in Part Family Modelling"*. Annals of the CIRP. Vol. **43/1**, pp. 129-132. **1994**.

Mäntylä, M., D. Nau, and J. Shah. "*Challenges in Feature-based Manufacturing Research"*. Communications of the ACM. Vol. **39**(2), pp. 77-85. **1996**.

Nielsen, E. H., J. R. Dixon, and E. E. Zinsmeister. "*Capturing and using designer intent in a design-with-features system*". DTM'91: 3rd. Int. Conf. on Design Theory and Methodology (ASME-DE), Miami, Florida, USA. Sept 22-25, Vol. **31**, pp. 95-102. **1991**.

Nnaji, B. O., H.-C. Liu, and U. Rembold. "*A product modeller for discrete components*". International Journal of Production Research, Vol. **31**(9), pp. 2017-2044. **1993**.

Salomons, O. W., F. J. A. M. van Houten, and H. J. J. Kals. "*Review of Research in Feature-Based Design*". Journal of Manufacturing Systems, Vol. **12**(2), pp. 113-132. **1993**.

Shah, J. J., A. Ali, and M. T. Rogers. *"Investigation of Declarative Feature Modeling". (ASME) Computers in Engineering Conf. and Exhibition*, Minneapolis, Minnesota, USA, September, 11-14, Vol. **1**, pp. 1-11. **1994**.

Sheu, L.-C. and J. T. Tin. "*Representation scheme for defining and operating form features*". Computer Aided Design, Vol. **25**(6), pp. 333-347. **1993**.

Silva, R. E. d., K. L. Wood, and J. J. Bearman. "*Representing and Manipulating Interacting Interfeature Relationships in Engineering Design for Manufacture*". (ASME) Advances in Design Automation, Vol. **DE-Vol-32-1**, pp. 1-8. **1990.**

Sodhi, R. and J. U. Turner. "Representing Tolerance and Assembly Information in a Feature-Based Design Environment". (ASME) Advances in Design Automation Conf., Vol. **DE-32-1**, pp. 101-108. **1991**.

Stroud, I. "*Modelling Techniques for Handling Non-Geometric Information*". Second Symposium on Solid Modeling and Applications, May 19-21, Montreal, Canada, Vol. pp. 367-376. **1993.**

Taylor, L. E. and M. R. Henderson. "*Validating a feature-based meta-model for mechanical products: a case study*". In Advanced CAD/CAM Systems: State-of-the-art and future trends in feature technology, R. Soenen and G.L. Olling, Editor. Chapman & Hall (IFIP): London. pp. 219-239. **1996**.

Tomiyama, T. and P. J. W. ten Hagen. "*Representing Knowledge in two Distinct Description: Extensional vs. Intensional".* Artificial Intelligence in Engineering. Vol. **5**(1), pp. 23-32. **1990**.

Tomiyama, T., Y. Umeda, and H. Yoshikawa. "*A CAD for Funcional Design*". *Annals of the CIRP*, Vol. **42/1**, pp. 143-146. **1993**.

Vancza, J. and A. Markus. "*Features and the principle of locality in process planning*". International Journal of Computer Integrated Manufacturing, Vol. **6**(1-2), pp. 126-136. **1993**.

Yoshikawa, H. and K. Ando. "*Intelligent CAD in Manufacturing*". Annals of the CIRP, Vol. **36/1**, pp. 77-80. **1987**.

Zeid, I. "*CAD/CAM: Theory and Practice*". Computer Science Series, Vol. 1. McGraw Hill International Editions. **1991**.

Zhang, K. F. and H. H. Elmaraghy. "*Validity Check for a Functional-oriented Modeler*". (ASME) Advances in Design Automation, Vol. **DE-65-2**, pp. 293-300. **1993**.