A Layered Framework for Structuring Product Data:

The development of a product data model

Tim Murdoch

Cambridge Engineering Design Centre
Engineering Department
Trumpington Street
Cambridge
CB2 1PZ
UK

Tel: 44 - 1223 332673
Fax: 44 - 1223 332662
Email: tnsm@uk.ac.cam.eng

Summary

Supporting the capture, storage and presentation of design data is a vital research activity in today's increasingly electronic design office. Modern computing techniques are able to store large quantities of information from a wide range of sources. While this has led to the capture of important information, the current techniques of storage and presentation are relatively unstructured, badly indexed and poorly presented. These obstacles make it difficult for an individual to access the appropriate data, inhibiting its reuse and leading to the possible repetition of expensive and avoidable mistakes.

This position paper outlines the development of a dynamically layered framework for design data which supports both the design artefact (product being designed) and the design process (activities involved in designing the product). Though still in the early stages of development, the framework has been prototyped using object-oriented techniques and currently forms the product data structure for several EDC design tools. Using a common structure for very different design data, the tools are able to share data through a multi-user database environment. The paper discusses some of the key issues for structuring product data which have emerged in the development of the design tools and in collaboration with industrial partners.

Introduction

Traditional paper-based methods and CAD tools have focused on capturing the geometry of individual parts within a product. Information other than geometry, such as material, manufacturing process and weight, was added by annotating the drawings and files. The overall structure and versioning was provided by bills-of-materials and personal file systems. In contrast, modern CAD tools use parametric models, embedded constraints and database driven configuration tools to capture deeper levels of design data and create well-managed, persistent and version-controlled product structures. However, the indices used to structure the latest product data models are still based around the traditional bill-of-materials type lists of information, ie. Computer-Vision's Configuration Access Tool.

Against this, design research is showing that product information is often complex, highly interconnected and mostly non-geometric. Entity-relationship diagrams, successfully used in electrical and control systems design, are becoming increasing used to describe mechanical systems. This includes both the conceptual (Function Structures [Pahl & Beitz, 1985]) and the more detailed stages (Functional Modelling [Johnson, 1992]). As the underlying structures of the artefact have become better understood, so too have the activities of the design process. Both prescriptive [Pahl & Beitz, 1985] and descriptive [Blessing, 1994] models have been proposed and much work has been undertaken to record and understand design rationale [Ullman, 1991, Hales, 1988].

Within the Cambridge EDC numerous design theories have been proposed and several have been implemented. Each one operates on specific highly structured design information, and each one must be fully supported. These include:

- FUNCSION, functional synthesis [Chakrabarti, 1992],
- KATE, configuration optimisation [Murdoch, 1993], and
- PROSUS, design process support matrix [Blessing, 1994].

FUNCSION operates by assembling orthogonal functional elements according to functional compatibility (Figure). KATE operates on physical parameters to vary components within a given functional configuration in search for high quality solutions (Figure). Central to both, is the use of relationships (interfaces) between data entities (either elements or components). These combine to form a network of functionally and geometrically related entities. PROSUS uses matrices, made up of generic issues and activities, to log the actions of a designer against a product tree structure (Figure). Given the need to design an artefact, or sub-artefact, a single matrix is used to log activities such as the generation of concepts or the evaluation of costs.

The specification for a product data structure within the EDC is further complicated by needs from industrial partners. Several have recently *re-engineered* their design process in search of enhanced efficiency and effectiveness and so have developed their own data structures both for the artefact and for the design process (Figure). Added to this, many technical systems combine different technologies; the most obvious example being mechatronic systems. In such cases multi-disciplinary design teams are required each bringing a new perspective on the artefact (Figure).

These influences, plus those of other research groups, have led to the development of a general layered framework for structuring product data within the Cambridge EDC. This framework has been implemented in C++, supported by an object-oriented database. To date, six new design tools have been, or are being, built using this data structure and one translation device has been installed to import data from an existing Lisp-based design tool. Work is currently underway to support STEP AP203 Express data, IGES geometry files and NC machine code.

A Structure for Product Data

The EDC product data schema supports two highly inter-dependent, but distinct types of design data:

(1) a description of the artefact; what it is, how to make it, how to use it etc, and
(2) a description of the process by which the artefact was designed (the design rational); what were the key decisions, who was involved, what where the activities, what were the tools and so on.

Within these there are three basic tasks:

(1) Indexing,
(2) Storage, and
(3) Capture.

Indexing the Artefact

A Multi-Perspective View

The framework for indexing a single layer of artefact data is shown in the accompanying figures. In common with the traditional bill-of-materials approach to product data the *product* (ie a direct drive AGV) is broken down into *assemblies* (chassis) and *assemblies* into either *parts* (base plate) or further *sub-assemblies* (drive assembly).

Within the EDC artefact data structure, a *part* is broken down in terms of *components* (mounting flange) where the actual geometry is defined. The geometry of a *component* may be enhanced by the addition of *features* (drilled holes). *Interfaces* (base plate mounting flange - motor body) are shown between nodes on the product breakdown and describe the connections between artefact elements.

The product definition also lists a number of *systems* (pnuematic ot electrical) that provide new groupings of existing nodes in the product breakdown. *Systems* may be broken down into their constituents of *assemblies, parts* and *components* or further *systems* (gripper actuation).

The increased level of detail allows finer granularity and more accurate indexing of complex design data. Rather like software and control systems, this should lead to greater re-use of past design work and provides a potential for further research into case-based reasoning. The interface information is vital to modeling the functional and geometric interaction between adjacent objects. In the first instance this extends the parametric solid modelling capabilities to a whole configuration level and in the second it allows for a complete functional analysis of the artefact. When combined with multiple perspectives, it leads to clearer data dependency and ownership, and therefore provides the potential for better management of both data and people.

A (R)evolutionary View

The design of a product is often described as an evolutionary process, where the basic structure is adapted and enhanced as information becomes more concrete. However, this approach can lead to conflicts within a product structure and loss of early design information. In response to this, a non-evolutionary view has been adopted within the EDC.

The basic artefact schema has been described using examples from a mechanical breakdown of the product. Considered as simply a nodal framework, it may also be used to support other types of information, such as function, production or maintenance data. However, the nature of such types of information is often very different and each may have its own inherent structuring mechanism. The next figure demonstrates this. The first layer is shown supporting the functional description of the product and others are shown supporting embodiment, detail and life cycle information. Each layer uses a similar nodal structure to support different breakdowns of a single product, which may or may not be either formal or consistent, depending on the strategy being employed. While there may be clear causal links between, say, torque transfer functions, rotating shafts and bearing replacement schedules, some causal links may be indirect or remain unclear until after the design has been completed.

An example of the types of data that might be stored against various layers is shown in the next figure. Its similarities to the *chromosome model* are obvious [Mortensen & Andreasen, 1993]. Each layer in the model supports different types of information and potentially uses a full artefact breakdown structure. The links shown between layers map causal relationships between entities. The chromosome model, developed from a theory of domains [Andreasen, 1992], divides design data into four domains: Function, Organ, Assembly and Production. The view pursued within the EDC is that any number of layers could be created, depending on the requirements of the company, the complexities of the product and the design strategy being employed.

Alternative Artefacts

A number of alternative solutions to a given specification are often developed during the design process. Two mechanisms have been developed to support this. The first handles the comparison of completely different products to a customer specification (Figure). Each separate alternative is shown as a distinct multi-layered description of a product. The customer specification defining requirements for overall functionality, physical properties and in service use is stored separately and also as a multi-layered data model[1]. The second approach is to use standard database version-control mechanisms to handle changes made to either the nodal

[1] More detailed specification required during the design process are stored at artefact nodes as required.

structure or item of data within the artefact (Figure).

Indexing the Process

The design matrix for an item in the product breakdown described earlier was based around generic activities listed against specific design issues. Similar ideas to this have been developed by industry where generic activities in the design process were listed against specific product nodes and attributes. The product data schema developed within the EDC attempts to support both these views by attaching a list of issues to each item in the product breakdown (Figure).

The PROSUS design matrix is supported by providing the appropriate name for each issue and attaching a specific data resource object to index the generic activities. The industry-based design process is supported by wrapping issues around appropriate product attributes and using other data resource items to index further process information.

It must be noted that PROSUS specifies a single artefact breakdown using a simple tree hierarchy. While this product structure supports this approach, work has yet to be done to match the design matrix to the multi-perspective, multi-layered artefact network required to fully describe the characteristics of a product throughout its life. A first prototype of PROSUS is being developed for a single layered product breakdown using simple tree hierarchy.

Storage of the Artefact

The product structure provides a rich indexing system for the design artefact and process data. Each node in the index is a reference to either an *artefact* object or an *issue* object which contain lists of embedded data objects. It is these embedded objects which hold the design data.

Many types of information may need to be stored against the artefact including: functions, parameters, attributes, position and orientation in space, degrees of freedom between parts and components, geometry, production process. The types of data that are currently being stored on the data base fall into two categories:
(1) factual data, and
(2) physical and technical knowledge.

The factual data can be regarded as the information required to describe the way a product is: ie. the torque, diameter or weight of a shaft. The physical and technical knowledge is the information required to establish factual data: ie the maximum stress in a shaft with a specified diameter and wall thickness and known peak torque.

Traditionally factual data has been stored using lists of *attribute-value pairs* (ie. cost = £3.23) which is a simple and effective approach. The representation of physical and technical knowledge is an area of specific interest for both functional modelling and configuration optimisation and requires more sophisticated data storage requirements. Two of the first data objects to be defined were *Characteristic* and *Constraint* (Figure).

A Characteristic object holds lists of both functional and physical factual information. In this case the definition of function has been derived from the theories embodied in FUNCSION. The physical information has been divided into parameters and attributes; parameters being things that a designer may change (ie. length, bearing number or material), and attributes being things that a designer may affect (ie. weight or cost). Functions act at specific points-of-interest which require both positional and orientation information. A local co-ordinate system is also defined.

Constraints are a simple way of capturing shallow physical and technical knowledge. They relate factual data using equations and inequalities. By testing a constraint for consistency, one can establish whether some physical or technical constraint has been met.

The capture of further physical and technical knowledge has been supported by introducing parametric modelling techniques into the parameter and attribute data types. This allows their values to be computed from embedded equations and associated look-up tables enabling the designer or optimisation engine to study the full effects of design changes.

Storage of the Process

Data types for the design process are still being developed. However, one central data item is *event*. An event records and indexes a specific design action. It is indexed according to issue and activity, and also artefact node. It will also need to list artefact data which has been created or edited (already held on the artefact breakdown). Added to these it will need to list many other data items such as: what was done by the designer, what tools were used, process specific data, what supporting information was gathered, how long the event took, how many involved, what event preceded it, what event triggered the action, what were the key decisions and so on.

Capture of the Artefact

Specific design tools are required to capture product information. So far six design tools have been built around the data structure, five of which concentrate solely on the artefact.

- TreeView: The artefact structure and design matrices
- BuildSite: The artefact configuration and factual data
- CompGeom: Parametric geometry
- CompDef: Knowledge-capture
- KATE II: Configuration optimisation tool built around BuildSite and CompDef

Added to these is FUNCSION, the functional modelling tool, which has been linked by a translator for importing data. The experience being gained in using and supporting these tools on a wide range of design projects is providing important validation of the multi-layered, multi-presepective artefact structure and is contributing to the development of new data objects.

Capture of the Process

A process-based design tool is also being developed. Called PROSUS, it extends the

TreeView tool so that specific design events can be recorded against the design matrix issues. Currently using a single layered approach, it will not provide the opportunity to investigate the capture and recognition of causal links. This aspect will addressed in future work.

Conclusions and Future Work

A rich and dynamic product data model schema is being developed in the Cambridge EDC. Based on a dynamically layered framework the schema has been developed to satisfy technical requirements developed from the research within the EDC and experience in industry.

The schema provides a number of new aspects including:
(1) Multiple perspectives supporting concurrent design practice
(2) Multiple layers supporting many different types of information
(3) Multiple alternatives supporting top level version control and handling of customer requirements, and
(4) Integrated process schema supporting various matrix-based models of the design process.

Future work will address problems of:
(1) Implementing the PROSUS design process management tool
(2) Capturing causal-links between product layers, and
(3) Integration with STEP AP203 and potential development of a new application protocol.

References

Andreasen M.M., 1992, *The Theory of Domains*, Function Workshop, Cambridge University, D Ullman et al Ed

Blessing B., 1994, *A Process-Based Approach to Computer-Supported Engineering Design*, PhD Thesis, Twente University, Denmark

Chakrabarti A., 1992, *Designing by Functions*, PhD Thesis, Cambridge University

Johnson A.L., 1992, *Supporting Creativity with Computers*, Function Workshop, Cambridge University, D Ullman et al Ed

Mortensen N.H., and Andreasen M.M., 1993, *Structuring Product Data - based on the Chromosome model*. Technical Report, Technical University of Denmark.

Murdoch T.N.S., 1993, *Configuration Evaluation and Optimisation of Technical Systems*, PhD Thesis, Cambridge University

Murdoch, TNS & Ball, NR, *The Development of an EDC Product Data Model*, Technical Report, CUED/C-EDC/TR21, December 1994

Murdoch, TNS & Ball, NR, *A Layered Framework for Sharing Product Data*, ICED'95, Praha, August 1995

Pahl G., and Beitz W., 1988, *Engineering Design*, The Design Council, London

Staufer L.A., and Ullman D., 1991, *Fundamental Processes of Mechanical Designers Based on Emprical Data*, Journal of Engineering Design, Vol 2 No 1 pp 89-101

A Layered Framework for Structuring Product Data

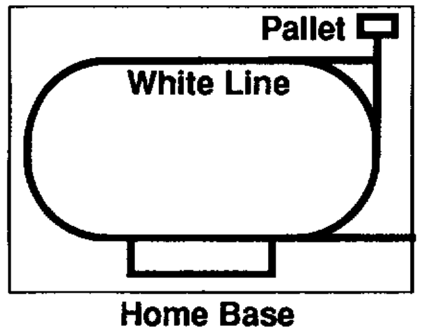
The development of a product data model

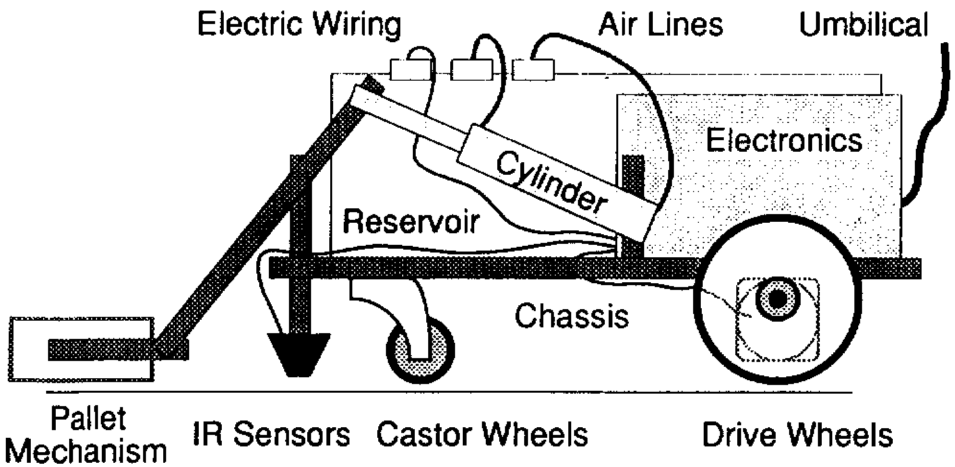
Tim Murdoch

A Layered Framework for Structuring Product Data

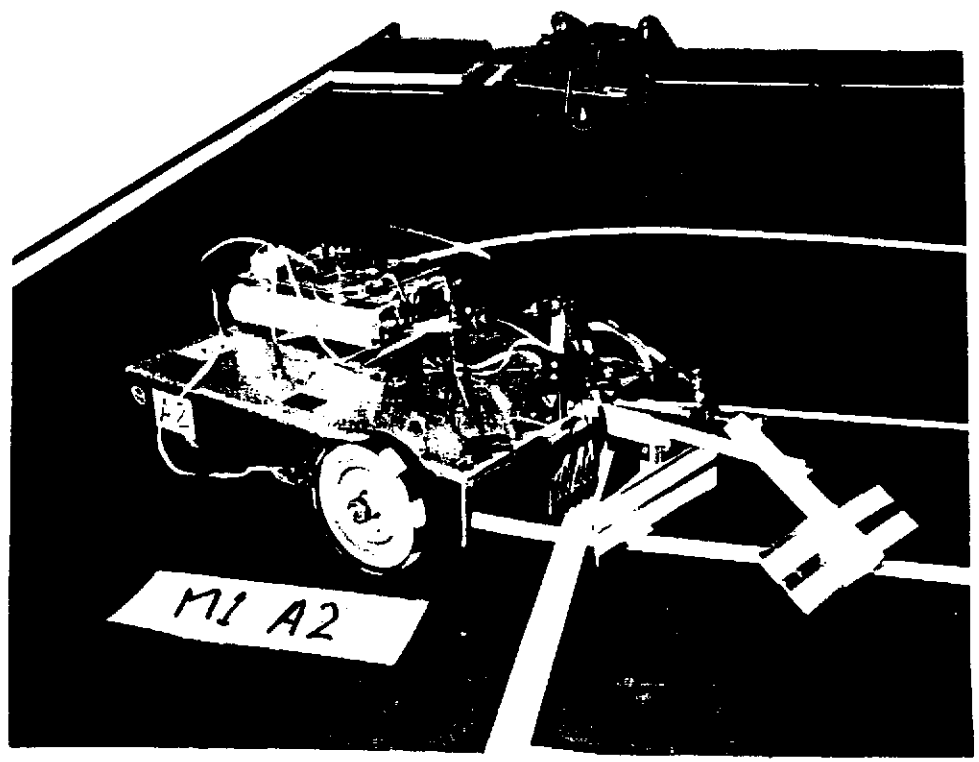
- Influences
- Existing Product Data
- Proposed Structure
- Artefact
- Process
- Conclusions and Future Work

Autonomous Guided Vehicle

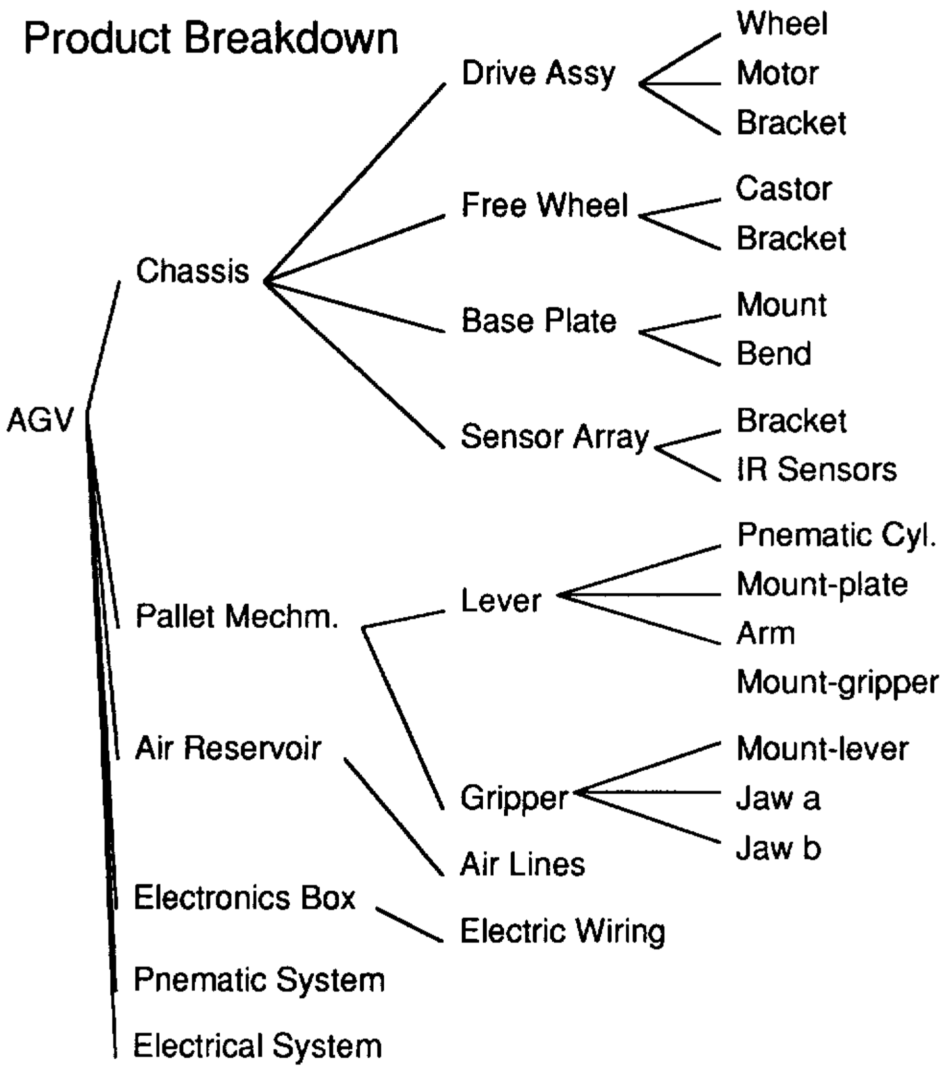




Autonomous Guided Vehicle

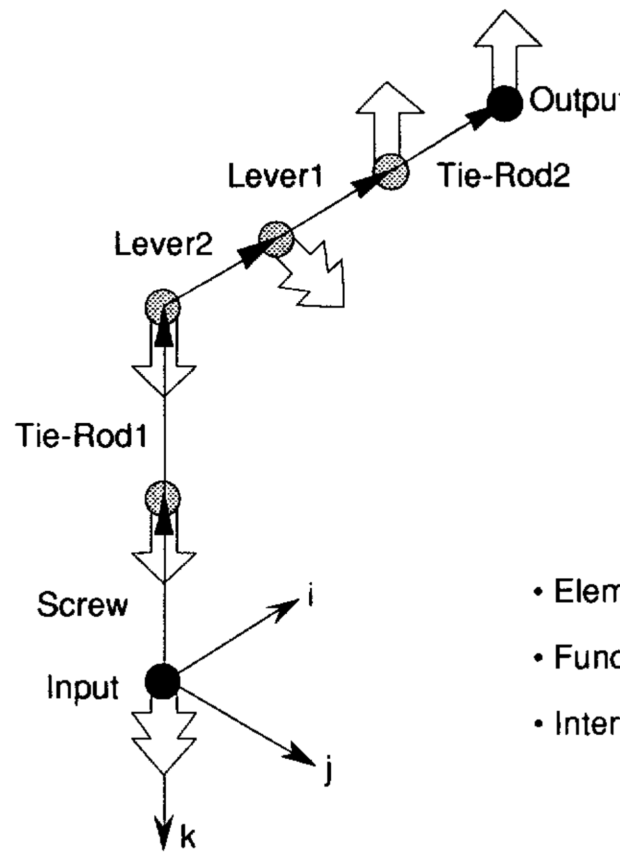


Product Breakdown



CAMBRIDGE EdC

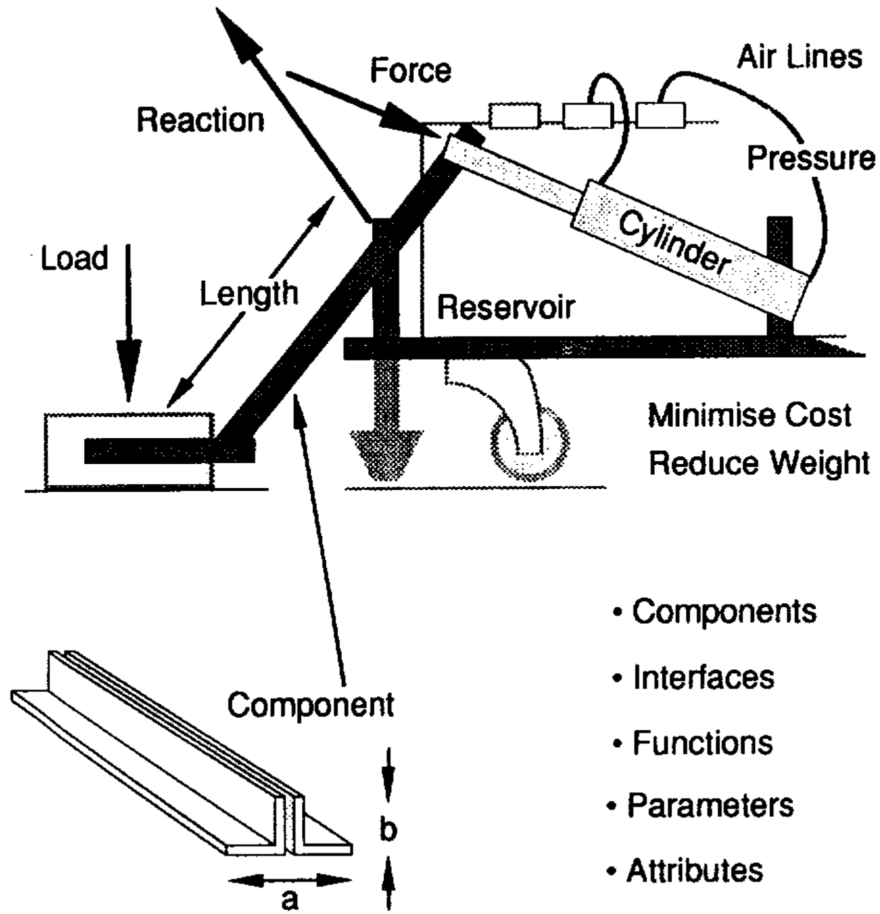
FUNCSION



- Elements
- Functions
- Interfaces

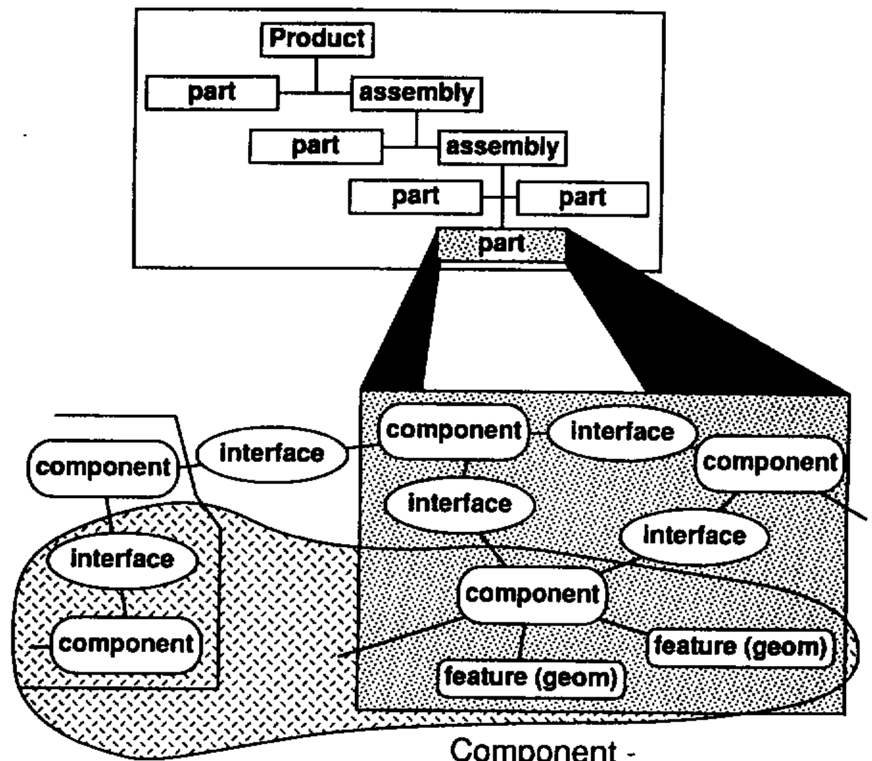
Tim Murdoch

KATE

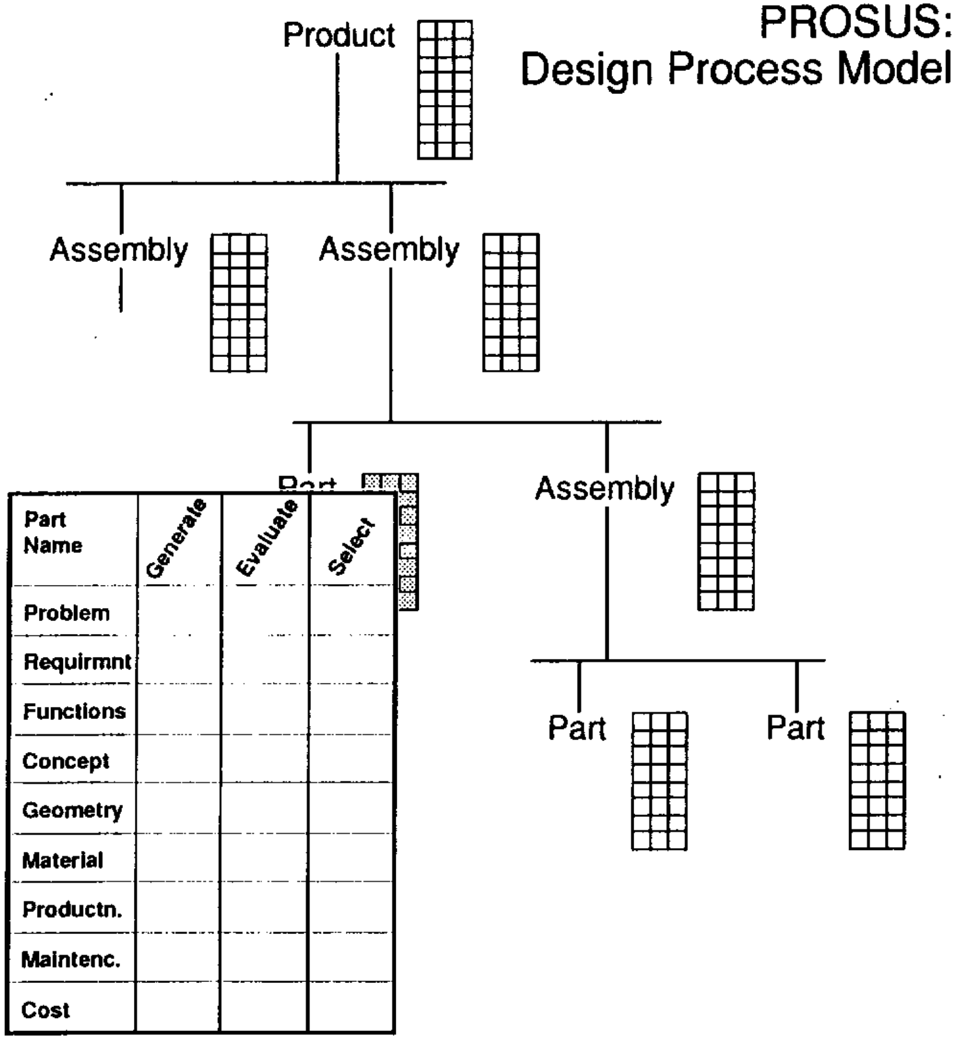


- Components
- Interfaces
- Functions
- Parameters
- Attributes

Product Structure



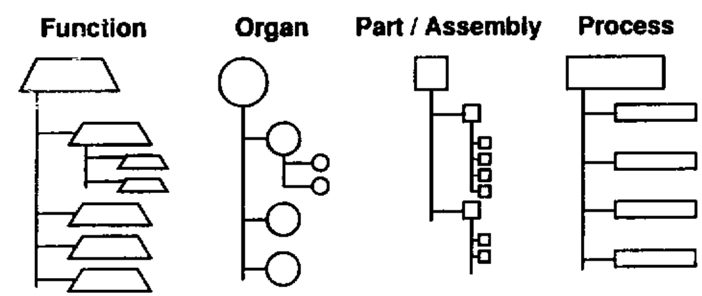
PROSUS: Design Process Model



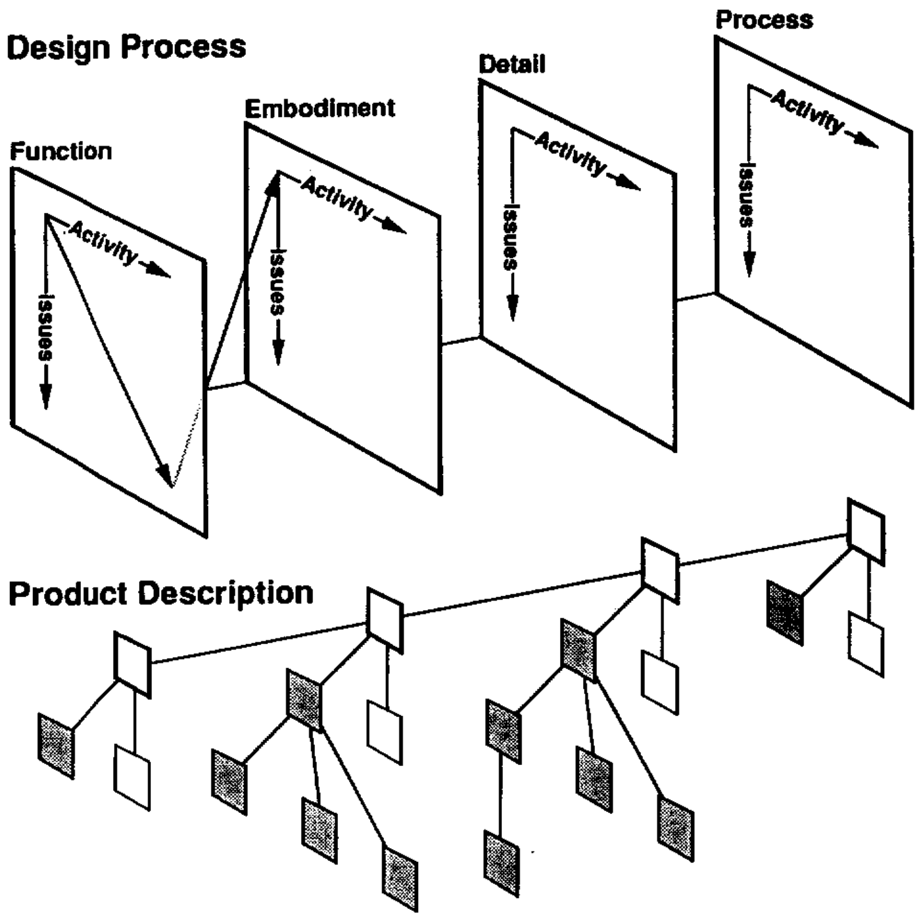
Domains Proposed for Design

French / Pahl & Beitz	Suh	Andreasen
Need	**Consumer**	**Function**
Concept	**Function**	**Organ**
Embodiment	**Physical**	**Part / Assembly**
Detail	**Process**	**Process**

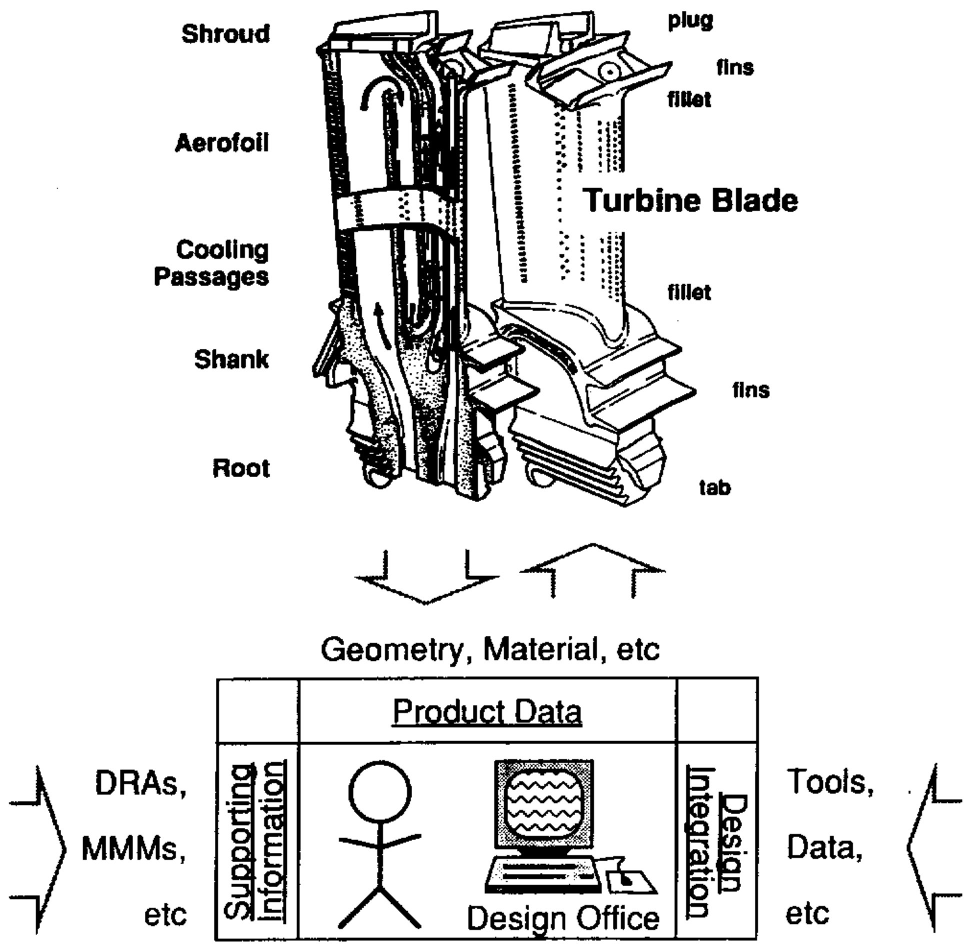
Andreasen's Domains - linked by "chromosome"

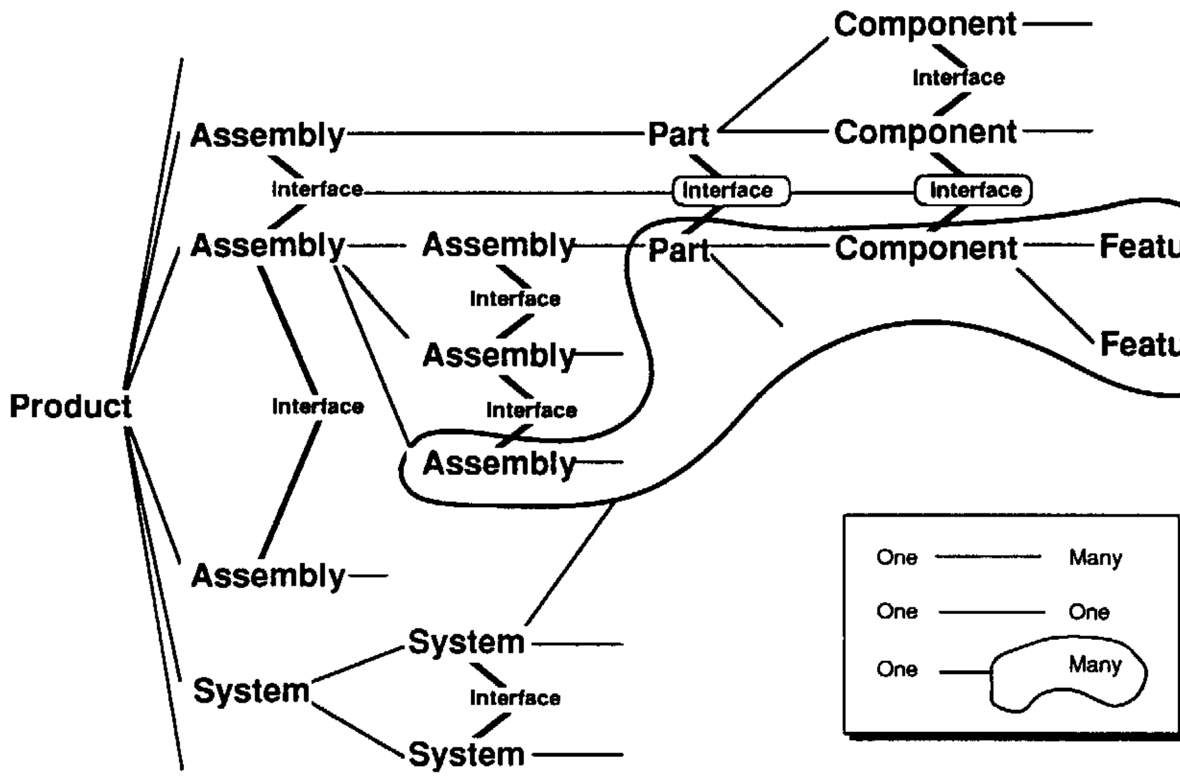


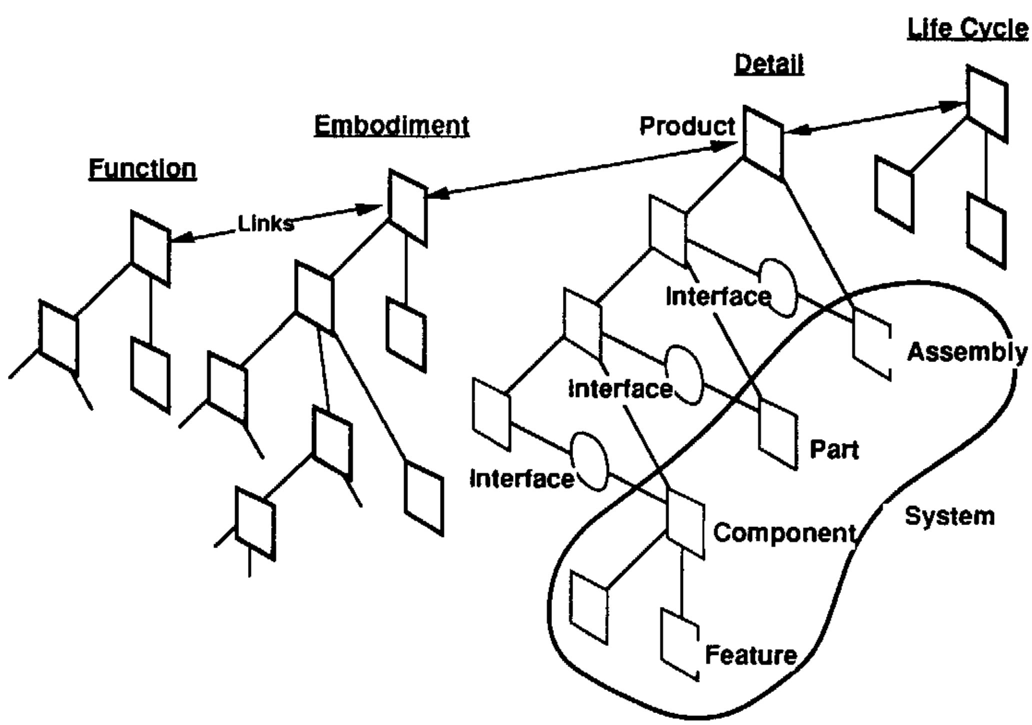
Product Data Context



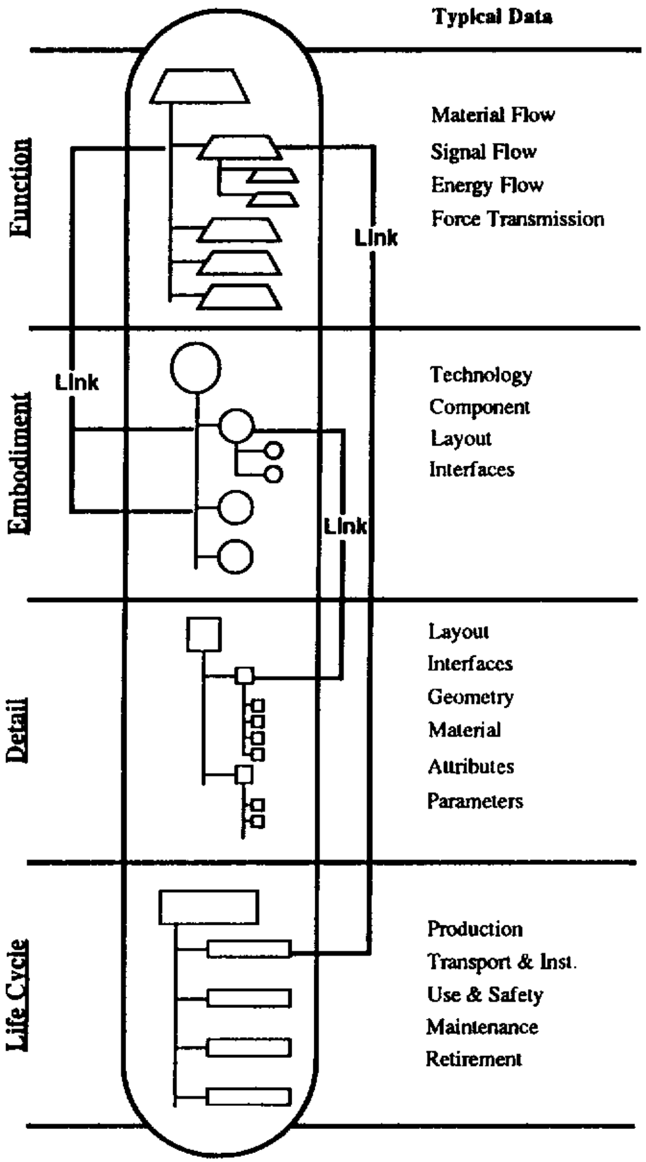
Design Integration







Four Example "Layers" from the Artefact Structure



Four Example "Layers" From the Artefact Schema

CAMBRIDGE EdC

An Index for the Design Process

Generic Design Process

Design Issues — Generate — Evaluate — Select

- Problem
- Requirements
- Function
- Concept
- Interfaces
- Layout
- Assembly
- Transport & Installation
- Maintenance
- Cost

Possible Index

Tim Murdoch

September 1994

CAMBRIDGE EdC

Product Based Design Process

Product Attributes — Evaluate Options — Plan Option — Realise Option

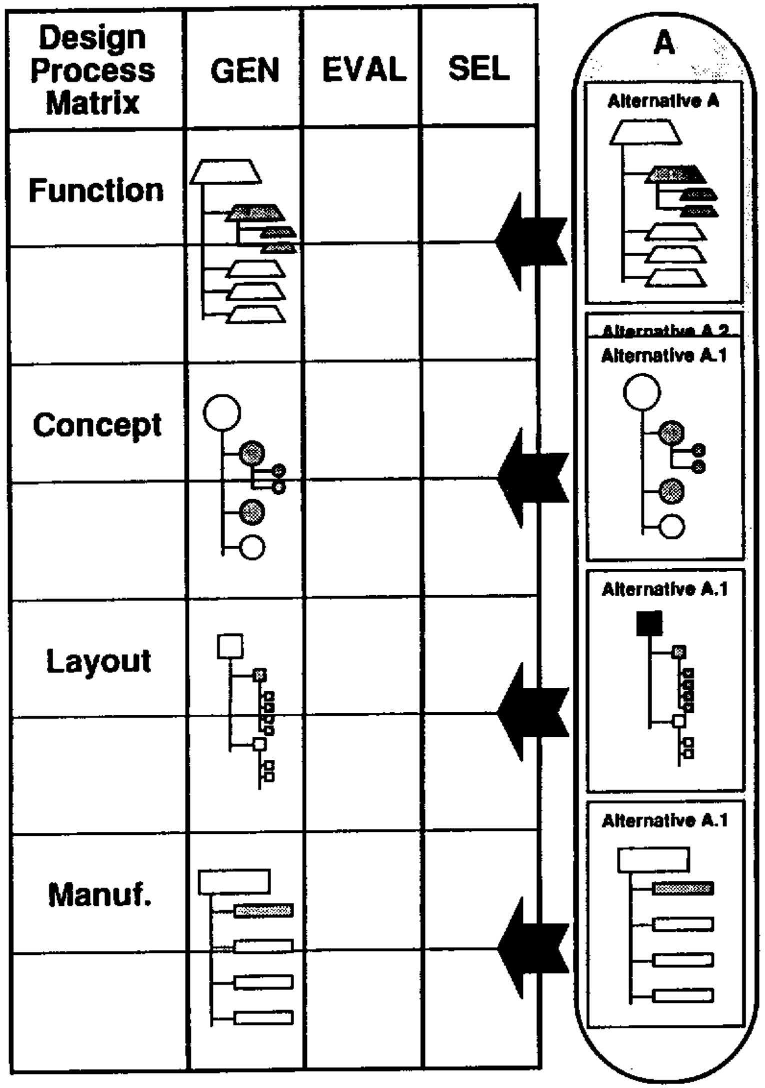
Functional
- Function / Performance
- Operation Capability
- Interfacing

Physical
- Layout / Concept
- Geometry
- Material
- Structural Properties
- Mass Properties

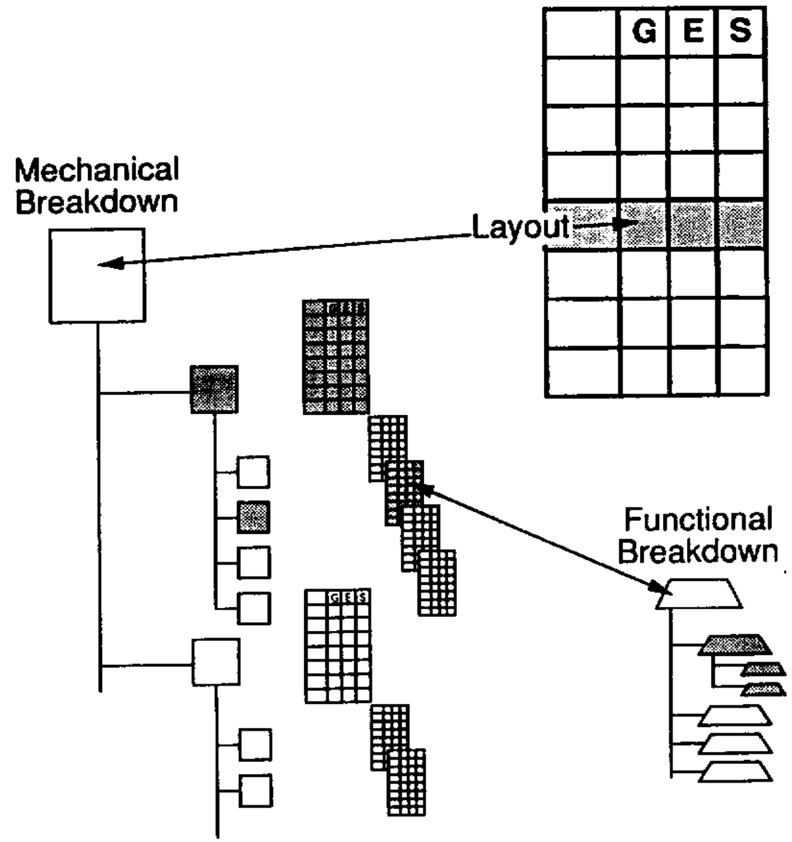
Life Cycle
- Operating Cost
- Safety & Reliability
- Emissions
- Manufacture
- Unit Cost

Possible Index

Tim Murdoch



Linking Artefact To Process



Conclusions and Future Work

- Multi-perspective
- Multi-layered
- Supporting range of design process strategies
- Integrated artefact & process structure
- Implemented on Six Design Tools

- Development of Integrated Design Framework
- Investigation of causal links
- Integration with STEP
- Links to commercial design software

CAMBRIDGE EdC

Tim Murdoch