FEMOD : A feature based modeller architecture

*Frédéric DAUGY¹, Frédéric FERU²
for The FEMOD CONSORTIUM*

*¹Aerospatiale, Joint Research Center DCR/I
92152 Suresnes , France*

*²University of Valenciennes - LAMIH (UA CNRS - 1775)
59 304 Valenciennes Cedex, France*

FEMOD, a Brite-Euram project, brought together most of the European aerospace companies and 3 well known laboratories around the goal of specifying a feature based modelling system.

The system, mainly dedicated to design and manufacturing engineering of milled parts and sheet-metal parts, will support new methodologies allowing dramatic productivity, and quality improvements.

Vertical integration of software applications from design to manufacturing around common product models built with features, allowing computerization of knowledge are the main concepts underlying these methodologies.

This chapter presents the architecture designed during the FEMOD project to answer user requirements induced by these new methodologies.

Basically, the architecture is built on three environments: the "Product Development Environment", the "Data Dictionary & Knowledge Handling Environment" and the "Software Development Environment". Based on an object-oriented approach, these environments cover the needs of the different users of the system from product designers to software developers.

The architecture is described at a functional level where functionalities, that the system must provide, are organised into components. It especially takes into account the system evolutivity, and customizability, and the ability to integrate off the shelf components.

The architecture highlights the considerable need not only for data standards, such as most of STEP parts, but also for functional standards such as SDAI or AIS (proposed by CAM-I) , in order to reinforce the modularity and evolutivity of the system.

Finally, the way forward to an extensive use of the FEMOD concepts is foreseen as a step-by-step evolution inside a long term strategic framework based on the

architecture described above.

15.1. INTRODUCTION

FEMOD (Acronym for “app^lication of FEature based MODe^lling for complex product design and manufacture”) was a Brite Euram project characterized by its “user-driven approach”. Most of the European aerospace companies and 3 well known laboratories were involved in this project from February 1991 to December 1993 to specify the user-requirements for a future feature based modelling system.

This project offered to partners an excellent occasion to discuss and exchange ideas. From this intensive process resulted a consensus on specifications for a future CAD/CAM system.

These specifications depict what users expect from the new generations of CAD/CAM systems and especially highlight the integration of company’s know-how in this system.

The specifications are now used as a basis for software evaluation and requests upon Information Technology (IT) vendors, to force them to offer softwares that better fit industrial needs.

After an overview of the project, this article presents the underlying concepts and the work performed on the architecture of the FEMOD system.

15.2. OVERALL PRESENTATION OF THE FEMOD PROJECT

15.2.1. The consortium

DASSAULT AVIATION¹ (France) was the project leader ; the other industrial partners were AEROSPATIALE (France), ALENIA (Italy), BRITISH AEROSPACE (United Kingdom), CASA (Spain), DASA (Germany), FOKKER (Netherlands), SAAB (Sweden). Universities and research centers were also involved : CNR-IMU (Italy), IVF-KTH (Sweden) and University of Valenciennes LAMIH (France).

The FEMOD consortium was large and consequently very representative of European aerospace industry.

The research centers brought in an external long term view and ensured a sufficient level of innovation.

15.2.2. Project goals

European aircraft manufacturers have been major CAD/CAM users for the last 20 years. They started from 2D geometry, then, migrated to 3D surface modellers. Currently, they are investigating 3D solid modelling in order to work on digital mock-ups of aircraft parts and assemblies.

¹ Prime contractor : DASSAULT AVIATION

Contact : J. PECHAUD - DGOI/ST-CFAO - BP12 - 78141 Vélizy
Villacoublay-FRANCE

In parallel with the evolution of CAD/CAM tools, methodologies of work concerning both the design phases and the manufacturing operations have deeply evolved to integrate the advantages of new techniques.

However, several problems still exist. They are caused by :

- 1 the low level of the entities manipulated by software, which are often pure geometric entities without engineering meaning,
- 2 the subsequent lack of information in CAD/CAM models, such as tolerances, design intents,
- 3 poor modification functionalities offered by most CAD/CAM systems,
- 4 difficulties to automate functions and computerize know-how,
- 5 the inability of softwares to fit well with internal company methodologies, instead of dictating them...

The FEMOD goal was then to elaborate IT supported methodologies enabling users to work more efficiently, in term of costs, delays and quality from design offices to shop floors. In parallel, FEMOD produced functional specifications for a feature based modelling system which will be the basis for a direct application of the defined methodologies.

The project scope was primarily focused on engineering and manufacturing engineering of milled parts and sheet-metal parts.

15.2.3. Project organization

The project was decomposed into 4 major steps. The first one consisted of state of the art studies about the use of CAD/CAM among the different industrial partners, researches performed on features, and commercial offers related to features and knowledge based systems [COQ 92] [FER 92b] [JON 93].

Then, a large effort was devoted to the definition of new methodologies. The third step illustrated new methodologies with software prototypes. Finally, the functional specifications for the future FEMOD system [DAU 93] and a generic product model framework were elaborated [HAR 94].

15.3. NEW METHODOLOGIES

15.3.1. Principles

With competition so keen, new methodologies aim at improving productivity and product quality in order to deliver a customer driven product at the right time, price and quality.

It is commonly admitted that about 75% of the price of a product is dictated by decisions made during design phases. Consequently, FEMOD made a special effort upon this phase. The following action guidelines were identified to reach our global objectives [JOH 92] [BOZ 93].

- 1 Human interpretation and multiple inputs of information already recorded, have to be avoided, e.g. manual creation of geometry needed for FEM.

- 2 Repetitive and low level tasks, e.g. geometric modifications, must be alleviated by automation, so that designers can concentrate on "pure mechanical work".
- 3 Any relevant information must be accessible so that users can make sound decisions. Decisions, that are incompatible with downstream constraints must be avoided, e.g. hole diameter that do not correspond to an available drill.
- 4 The later a problem is detected, the more expensive is its solution. Methodologies and mechanisms enabling people to detect potential problems are therefore required e.g. manufacturing simulation function...

All these actions contribute to reduce iterations and therefore the development cycle and the cost of the product. Moreover, a clear understanding of the whole product development process leads to the identification of problem sources, and outdated practices, and, finally to an improved quality of the product and its associated process.

15.3.2. Enabling techniques

Apart from the organizational and business aspects, the actions listed above must be largely supported by software tools to be successfully applied. The current CAD/CAM tools are not sufficient for that, and some new complementary IT techniques are then essential to improve them.

15.3.2.1. Product models and features

Product models and features are fundamental concepts to get "intelligent information carriers" from product specification to quality control [JOH 92] [BOZ 93] [SHA 89] [WAR 92]. They allow the modelling and storing of any kind of information related to the product in such a way that it becomes easily accessible and interpretable by software applications and human beings.

A broad and open definition of feature was adopted for FEMOD : "A recurring pattern of information related to product descriptions". According to this definition, features are very similar to the notion of IT objects. While features are often context dependent, they are classified into categories : specification, design, manufacturing features, etc (e.g. an assembly constraint is a specification feature, a fixing hole is a design feature and a slot or a pocket are manufacturing features).

In the FEMOD approach, features are structured into product models. Product models link features together, taking into account dependencies, such as mapping of design features into manufacturing features. So, it describes the structure, shape and function of the product.

As product models are also context dependent, the notion of Total Product Model has been introduced as the whole set of information used to describe a product during its entire life cycle. So, it contains other models derived from the product model (e.g. FEM models, tools models which describe the necessary tools,...).

Product models and features are built according to predefined structures which are called Generic Product Models and Generic Features Definition [HAR 94]. These generic structures are handled by classes and relations in an object-oriented way.

In order to store and manage all the created classes, the notion of Data Dictionary was introduced [BRU 92] [GIE 92] [JOH 92] [HAR 94].

This approach differs from the one used in many recent CAD/CAM tools. In these tools, the notion of feature simply extends the geometric Boolean operations. Features are implemented as isolated geometric constructs directly inserted into a geometric model.

15.3.2.2. Integration techniques

By integration, we mean sharing of the same computerized information, and avoiding of duplications.

Integration eliminates multiple acquisition of the same information, hazardous data conversion, human "reinterpretation"... Maintaining total product models consistency is therefore much easier [KOS 92].

As aerospace projects are often based on complex, multi-site organizations, integration techniques are also fundamental to promote teamwork and sharing of information in a distributed environment [CAL 89]. Many integration techniques can be found in advanced database management systems including powerful modelling capabilities, access controls, distribution, versioning...

15.3.2.3. Knowledge formalization & computerization

To cleverly assist users, propose solutions, anticipate the results of a decision, check its validity, the IT system needs computerized knowledge. If classical programming languages are able to implement some forms of knowledge, many limitations exist especially in terms of maintainability and readability of this knowledge.

Indeed, knowledge elicitation will probably be an endless process because of the huge amount of knowledge referred by design and manufacturing activities, and its permanent evolution [BOZ 92]. These reasons increase the needs for perennality, evolutivity and maintainability of the computerized knowledge, which are not fulfilled by classical programming approaches.

Techniques adapted from artificial intelligence offer better functionalities to help formalizing, computer processing and maintaining this knowledge, on a company-wide basis.

Knowledge may be related to products (e.g. calculation formula for a web thickness), processes (e.g. list of operations to manufacture a pocket), or resources (e.g. list of available drills). So, it can be decomposed and represented through several forms:

- 1 formula,
- 2 production rules,
- 3 tables,
- 4 sketches,
- 5 scenarii,
- 6 unformal (free text...)
- etc...

Knowledge can then be used according to different ways which can be divided in three categories corresponding to an increasing complexity : consultation,

verification and generation.

- 1 Consultation corresponds to a context dependent on-line help or information retrieval from the Total Product Model
- 2 Verification consists in validating a proposed solution.
- 3 Generation consists in using knowledge to propose a solution.

These few remarks show the difficulty of knowledge formalization, and implementation. Open issues have even not been totally addressed yet by AI techniques. Consequently, a pragmatic approach was adopted. The system should provide a repository to handle computerized knowledge called the knowledge handler ; but only consultative knowledge use is mandatory.

More advanced uses of knowledge are possible by using inference engine mechanism, constraint propagation... However, they have to be performed on a local basis, to preserve the genericity of the architecture.

15.3.2.4. Automation

Several steps of the product life cycle can be automated in order to free users from repetitive or low-level task such as geometry manipulations... However, automation can also be more ambitious when trying to automate process planning for example.

Automation has to be managed wisely, otherwise it can be very costly and sometimes unusable while not flexible enough. It is then very important to give to users the freedom to amend the results generated automatically, without breaking the flow of information.

Product models and features facilitate automation because of the richer product description ; recognition or interpretation of information, often needed when working on pure geometric models, are eliminated.

A key issue is that automation involves knowledge. "Hard-coding" often required for automation efficiency, and the needs concerning knowledge already mentioned might be difficult to combine.

15.4. OVERVIEW OF FEMOD ARCHITECTURE

15.4.1. Approach for describing architecture

The previous considerations led us to define an architecture answering the industrial needs, independently of any specific tools.

Two important constraints were :

- 1 to design an architecture open enough to allow the widest possible use of off the shelf components,
- 2 to allow easy evolutions towards the ultimate level of functionality, and customizations to the specific context of each company.

Three levels of architecture description were introduced : functional, logical and physical levels :

- 1 the functional level organizes the functionalities available in the system into software components. The different functions to be fulfilled by the system are described here
- 2 the logical level refines each component by describing the objects it manipulates and interfaces between components. It remains independent of any software or hardware
- 3 the physical level describes the real IT objects ; it is linked with implementation and subsequently depends on softwares and hardwares choices. It is therefore beyond the FEMOD scope.

Our purpose focuses on the functional level.

15.4.2. The 3 environments

The functional level of the FEMOD architecture is composed of 3 different environments (figure 1).

- 1 The "Product Development Environment" supports the product development itself (e.g. design, process planning) and product development administration (e.g. project management, configuration management...)
- 2 The "Data Dictionary and Knowledge Handling Environment" provides functions for a first level of customization, which should not require IT specialists.
- 3 The "Software Development Environment" contains tools for deeper evolutions and customizations, requiring programmers. It should be usable both by IT vendors and internal developers.

Figure 1. Overall view of the FEMOD architecture

These 3 environments share product models, a data dictionary containing class descriptions, knowledge bases storing knowledge, and high-level feature-based functionalities. These shared components reflect the four enabling techniques previously identified : product models and features, integration, knowledge formalization and computerization, and automation.

15.4.3. The Product Development environment

This environment provides end-users with an extensible and customizable set of high level automated and/or intelligent functionalities which allow to interact with product models. They mainly impact on design and manufacturing activities, but technical documentation, maintenance, industrial management,... can also take advantage of them.

Examples of such functionalities are :

- 1 verification of manufacturability during design,
- 2 local automation of design (e. g. calculation of a rivet diameter, calculation of a web thickness...),
- 3 automated design of manufacturing tools, fixtures or jigs,
- 4 automated process planning,
- 5 automated tool selection,
- 6 nesting,
- 7 coding & classification...

These functionalities are very dependent on the kind of parts and the activities considered. Functionalities needed to support the design of a wing rib are different from the ones required for manufacturing engineering a sheet-metal part. Figure 2 describes how a functionality, dealing with sheet-metal parts, could look like.

Figure 2. Apparearances of the product development environment

A "generic functionalities" list can be proposed (figure 3) :

- 1 **Product modelling functionalities:** They are used to create, query and modify product models. They are only data manipulation functionalities. However, they can trigger evaluation, validation or generation functionalities.
- 2 **Evaluation functionalities:** They allow to derive information from product models concerning criterias as cost, manufacturability, assemblability... and permits to verify the design consistency against the given specifications.

- 3 **Validation functionalities:** They are similar to daemons constantly verifying that a given criteria remains valid after an interaction with a product model. If it is not the case they prohibit this interaction.
- 4 **Generation functionalities:** These functionalities generate a solution automatically from some inputs. Following the distinction between generative and variant approaches in process planning, generation functionalities can be based on a "reasoning process" or an "analogic process".
- 5 **Administration functionalities:** They enable users to produce statistics, to classify product models...

Figure 3. Architecture of the product development environment

These functionalities support the product development cycle. They are based on progressive reasoning, catalogue selection, creation and modification specific mechanisms [FER 93].

These advanced functionalities also need the services of lower level ones for :

- 1 visualization of all created entities, relation and attributes of a feature, links between different features,...
- 2 user-interface,
- 3 dialogue management that provides users with a user-interface dedicated to each design phases. This functionality is already largely available in windowing systems (e.g. Motif), however, they can gain in intelligence, as the user's intent will be known by the software. For example, it is easier to highlight a pocket and to display context dependent menus and help when a

- pocket is selected...
- 4 interfaces with storage and external geometry unities in order to permit the use of different geometric modellers (e.g. design, stress analysis,...) or alternative geometry representations (solid model, surface model,...) needed by different applications

15.4.4. The Data Dictionary and Knowledge Handling Environment

The objective of this environment is to enable a first level of customization, accessible to persons who have a mechanical background but are not IT specialists.

Examples of such customizations are :

- 1 insertion of a new attribute in a class,
- 2 modification of a parameter (e.g. the maximum diameter of hole a machine can drill),
- 3 modification of some kinds of knowledge pieces such as unformal explanations, tables,...

The "Data Dictionary and Knowledge Handling Environment" contains functions for creating, storing, managing and maintaining generic models i.e. classes, and knowledge pieces. This environment also supports the general system administration. It is built of two main components : the Data Dictionary Handler (DDH) and the Knowledge Handler (KH).

15.4.4.1. The Data Dictionary Handler (Figure 4)

From conceptual class schemes of features, the DDH is able to derive a feature class definition, and store it in the data dictionary. Functionalities to manage class definitions are also provided.

Feature class definitions contained in the data dictionary can then be used for automatically generating product model database schemes, data access and manipulation routines, user-interface buttons and menus and geometric manipulation routines. These routines are helpful for accelerating software development, within the Software Development Environment.

Feature class definitions can finally be queried from the Product Development Environment to inform end-users on the structure of features, required inputs...

15.4.4.2. The Knowledge Handler (Figure 4)

The second component of this environment is the Knowledge Handler, which allows to build and manage Knowledge Bases. Knowledge bases refer to class definitions of the data dictionary. The consistency between the classes described in the data dictionary, and the references to classes made from knowledge bases is guaranteed by special mechanisms.

As the KH is to be used on a company wide basis, mechanisms are needed to efficiently manage the effectivity of knowledge : a rule may be applicable only in the case of a military aircraft...

Many different forms of knowledge have to be supported : rules, formulas,

scenarii, sketch, unformal explanations, tables, libraries of components, existing validated parts...

Figure 4. Uses of the Data Dictionary and the Knowledge Handler

The underlying idea of the KH consists in "reifying" the knowledge, i.e. transforming knowledge into a collection of objects, the knowledge pieces. A knowledge piece is an instance of a class describing a form of knowledge mentioned above. A taxonomy of these classes is given in figure 5.

Figure 5. The Knowledge piece class hierarchy

The need for a stable and standardized knowledge description is crucial for companies. Following the "knowledge reification", the EXPRESS language from STEP-ISO 10303 was proposed to describe knowledge piece classes.

Below are two examples of a generic knowledge piece, and an inequality, both described in an EXPRESS-like way :

```

ENTITY KnowledgePiece;
  SUPERTYPE OF (ONEOF (
    Formula
    LogicalRule
    Table
    LibraryStandardComponents
    ExistingValidatedPart
    InformalExplanation
    Scenario
    Sketch
    Combination )) ;

  title : STRING ;
  author : STRING ;
  date : STRING ;
  version : INTEGER ;
  domain : KPDomain ;
  source : KPSource ;
  effectivity : SET [0, ?] CaseDescription ;
  conditioned-by : SET [0, ?] KnowledgePiece ;
  precedence : SET [0, ?] KnowledgePiece ;
  DDReferences = SET [0, ?] DataDictionaryReference ;
  related-internal-instructions : SET OF STRING ;
INVERSE
  is-preceded-by : SET [0, ?] OF KnowledgePiece FOR precedence;
END_ENTITY;
ENTITY Inequality ;
  SUBTYPE OF ( Formula);
  variables = LIST OF LocalVariable ;
  strict = BOOLEAN ;
  right-exp = MathExpression ;
  left-exp = MathExpression ;
END_ENTITY;
```

15.4.5. The Software Development Environment (Figure 6)

Nevertheless, the first level of customization described above is too limited to allow all customizations and evolutions. The Software Development Environment supports this "deep level".

This environment is dedicated to persons who are familiar with computer sciences. It makes it easy to enrich the Product Development Environment by :

- 1 developing new feature-based functionalities,
- 2 maintaining the existing functionalities,
- 3 interfacing and integrating system components.

It provides :

- 1 CASE functionalities to support the whole software development process (specification tools, data modelling tools, activity modelling tools, compilers, linkers, debuggers...)
- 2 libraries of shared resources, such as geometric calculation functions, user-interface and dialogue manager, product models and database access functions,...

The CASE functionalities rely on the data dictionary, the knowledge handler, and the shared resources, so that each piece of code developed in this environment uses them every time it is possible.

Figure 6. The Software development Environment

15.4.6. Standards and FEMOD system

Conformity with standards is essential. Indeed, each functionality of the Product Development Environment may evolve independently, especially if they are bought

from an IT supplier : e.g. FEM calculation software, geometric modelling tools... Standardization of data, and "interface functions" is therefore the only way for a company to integrate the state-of-the-art commercial tools and to guarantee the perennality of the whole system.

Among the standards to be considered, STEP-ISO 10303 plays a key role, while it will provide :

- 1 "integrated resources" for data representation (especially geometric data), which should be used as a basis to build the data dictionary,
- 2 a data description language EXPRESS (STEP part 11) which should be used as the data dictionary language, and could be used to describe knowledge,
- 3 a set of data access functions, SDAI, (STEP part 22) which should be proposed as a Software Development Environment resource to access the data dictionary and product models,
- 4 a series of Application Protocols, which could be implemented within the FEMOD system.

However, SDAI functions are only data access functions independent of the semantics. They need to be complemented by semantic dependent standards such as AIS [AIS 90]. AIS has been proposed by CAM-I as a set of standardized functions to operate on geometric entities, e.g. calculation of solid intersection...

15.5. CONCLUSIONS & PERSPECTIVES

Several short-term benefits already resulted from the project : the feature concepts are better understood by the partners who now have acute competencies in this field, relationships between industry and universities were reinforced, some mock-ups are being industrialized...

If many 'local' applications of the concepts are already possible within limited scopes, a direct company-wide application is still a long term goal. The reasons are some technical issues, the cost, and the lack of commercial offers for supporting some aspects of the system.

The FEMOD consortium believes that the industry will have the foresight to recognise the benefits of pursuing an active policy in this field. Companies will surely adopt progressive and incremental migration strategies to map short term actions and long term goals.

We hope that the software vendors will take advantage of the FEMOD achievements and provide products in line with these ideas, enabling the users to migrate more easily towards the FEMOD concepts. In doing so, vendors will increase their competitiveness and also release the industry from the need of extensive in-house software developments.

It is also our hope that the FEMOD project will influence the STEP organization, which is a fundamental standard for the FEMOD system.

All these contributions will aid the European industry in improving its product development by adopting the methodologies proposed by FEMOD.

15.6. REFERENCES

- [AIS 90] "Solid modelling - Application Interface Standard (AIS) 2.0", CAM-I report - Integrity Systems - 4/6/1990
- [BOZ 92] I. BOZZA, G. COLOMBO, U. CUGINI, I. VICINI, E. COQUEBERT, F. FERU "Knowledge extraction of sheet metal parts and milled parts on design and manufacture", FEMOD subtask 1.4.1 - technical report - 12/06/92
- [BOZ 93] I. BOZZA, U. CUGINI, F. MORONI, I. VICINI "Define new methodologies", FEMOD subtask 2.2 - technical report - 15/7/1993
- [BRU 92] W.J. de BRUIJN TNO Building & Construction Research "The IMPACT database management approach", 1992
- [CAL 89] OSD CALS Office & DACOM "Preliminary CALS Phase II architecture" D-779-89-01.2 - Office of the Assistant Secretary of Defense, Washington, D.C.
- [COC 92] E. COCQUEBERT, F. FERU, R. SOENEN "State of the art and evolution in feature-based modelling", Revue Internationale de CFAO et d'Infographie, Vol7, N2, '92
- [DAU 93] F. DAUGY "The FEMOD architecture", FEMOD subtask 4.1 - technical report - 23/11/1993
- [FER 92a] P.M. FERREIRA et. al. "Specification for a manufacturing planning enabling platform - Final report", CAM-I R-92-PPP/ANC/QAP-01 - University of Illinois at Urbana Champaign
- [FER 92b] F. FERU, E. COCQUEBERT, H. CHAOUCH, D. DENEUX, R. SOENEN "Feature-based modelling : state of the art and evolution" IFIP WG 5.2, WG 5.3, Tel Aviv, Israel, 13-15 April 1992
- [FER 93] F. FERU, C. VAT, A. TIMIMOUN, E. COCQUEBERT, C. ROUCHON "A design for manufacturing aided system based on functional and knowledge aspects", IFIP TC5/WG 5.3, TWCM '93, Phoenix, Arizona, 12-16 September 1993
- [FOW 89] J.E. FOWLER "A generic architecture for computer manufacturing software based on the product data exchange specification", NIST89-4168 - National Institute of Standards and Technology. 1989
- [GIE 92] W.F. GIELINGH & W.J. de BRUIJN TNO Building & Construction Res., J.R. Halbert Pafec Ltd "Implementation levels for semantic integration of open system CIM modules"
- [HAR 94] T. HARDY, B. PAPONOT "Specification of the information structure", FEMOD subtask 4.2 - technical report - January 94
- [JOH 92] J. JOHANSSON, R. WIKSTRÖM "IT4 Project, Integrated computer supported product development", SAAB Military Aircraft 1992 - internal report
- [JON 93] F. De JONG, R. HOOGEBOOM, T. REITSEMA "Survey on feature techniques parts 1 & 2", FEMOD subtask 1.3 - technical report - rev 3 - 12/2/1993
- [KOS 92] KOSANE K., VERNADAT F. "CIM-OSA : A Reference Architecture for CIM", Proceedings of the IFIP TC 5/WG 5.3 - 8th PROLAMAT '92 Conference, Man in CIM - Tokyo, Japan, 24-26 June 1992
- [MAG 92] S.P. MAGLEBY & K.B. GUNN "An object oriented environment for product modellers and applications", Brigham Young University USA
- [POU 90] J-C POUGNARD "Vers une architecture CIM", Proceedings of MICAD 90 - 9th International Conference - Paris, 12-16 February 1990
- [SHA 88] J.J. SHAH & M.T. ROGER "Expert form feature modelling shell", Computer Aided Design vol. 20 number 3 November 1988 Butterworth and Co LTD.
- [SHA 89] J. SHAH et. al. "Functional requirements of a feature based modelling system, Task 1 Preliminary report" CAM-I R-88-GM-04.1 - Arizona State University, Jan. 1989
- [WAR 92] E. WARMAN "The current architecture and reasons for its form", Design for Manufacture and Assembly Workshop -Leuven 19-20 November 1992