Technical Report
RAL-TR-96-098

Project MIDAS: Magnet Integrated Design and Analysis System

P Alotto U Barberis B Colyer C Greenough et al

December 1996

COUNCIL FOR THE CENTRAL LABORATORY OF THE RESEARCH COUNCILS

Project MIDAS: Magnet Integrated Design and Analysis System[1]

P Alotto[2], U. Barberis[3], B Colyer[4], C Greenough, T. Gutierrez[5], G. Jared[6], A Longo[5], P Molfino[2], G Molinari[2], E. Picco[3], J Simkin[4], N. Sormaz[6], D Thomas and C W Trowbridge[4]

Abstract

Results from a European collaborative project to create an open system for engineering design, with particular emphasis toward electromagnetic devices, are presented. The system, named MIDAS, is based on the new ISO STEP standard for data exchange and utilises state-of-the-art Graphical User Interfaces, Three-Dimensional Geometric Modelling and three-dimensional automatic mesh generation with an advanced two-dimensional subset.

---

[1]Current reports are available can be obtained using anonymous *ftp* from *www.inf.rl.ac.uk* in the directory */pub/mathsoft/docs*. This report is in file Midas4.ps.gz

[2]Dipartimento di Ingegneria Elettrica, Universita' di Genova, Italy
[3]Vector Fields Ltd, Oxford, England
[2]Ansaldo Ricerche CPI, Genova, Italy
[4]LABEIN, Bilbao, Espana
[5]Cranfield University, Bedford, England

Mathematical Software Group
Department of Computation and Information
Rutherford Appleton Laboratory
Chilton, DIDCOT
Oxfordshire OX11 0QX

Contents

1. Introduction

The main objective of the MIDAS project is the creation of an open system for engineering design [1]. The emerging International Standards Organisation (ISO) STEP (Standard for The Exchange of Product data) protocol provides the basis on which open systems can be built [2],[4]. Additional object classes required for the finite element analysis of electromagnetic problems have been defined within the project, thus the project has been involved in extending the standard itself. MIDAS was conceived as a multi-process client-server system, with a STEP compatible DataBase as the primary server via the STEP Data Access Interface (SDAI), see Figure 1. After defining the architectural features the paper deals with the components in turn i.e. the underlying structures based on STEP, the geometric modeller, automatic 3D mesh generator for finite element analysis and the advanced 2D subset and data exchange with third party systems.

2. The MIDAS environment

2.1 Architecture

The MIDAS Environment allows a user to execute design programs selected from a comprehensive set, either sequentially or concurrently. Each program may be provided with its own Graphical User Interface (GUI). All the programs are controlled by a single program, known as the Master Control Program (MCP); this has its own GUI by which means a design program may be started, paused, re-started or terminated. Each program which requires a GUI (including the MCP) comprises two processes; the first is an instance of a single GUI process, which acts as the user interface to the second process (or *application*). On a UNIX machine, the two processes communicate via UNIX sockets; all such communication is by text strings. This method was chosen because of its flexibility; many applications are developed initially to be driven by keyboard text input, and produce text output, so the task of converting them to being driven by text generated by a GUI is relatively sinple.

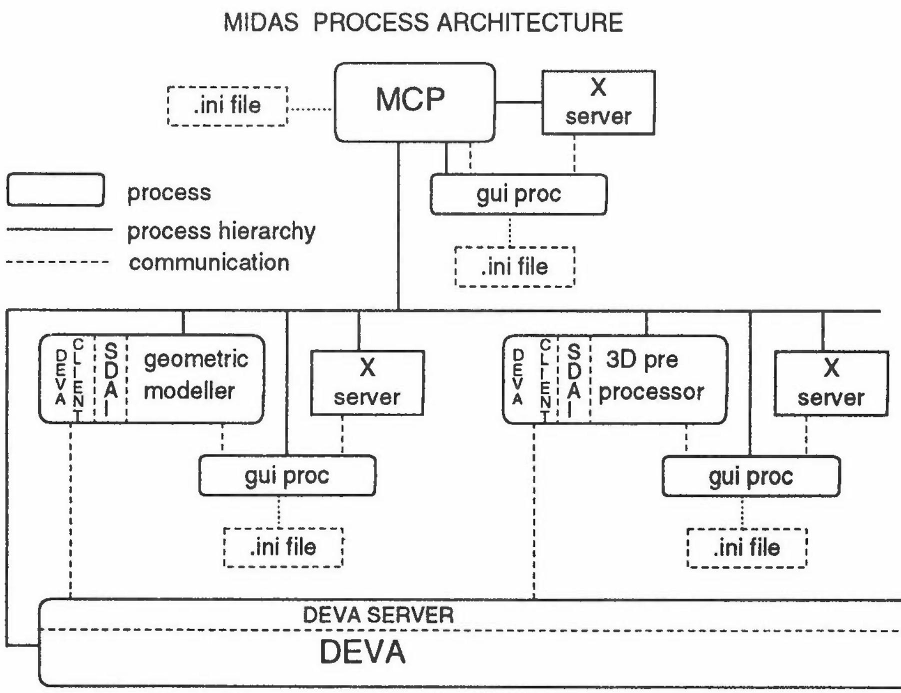

Figure1: MIDAS STEP Compatible Environment

When the MCP is started, it first reads ASCII data from a process initialisation file. This file provides for each process:

1. the initial dimensions and position of the window for the application and its GUI
2. the directory path of the GUI initialisation file
3. the directory path of the application's executable
4. names of sockets for inter-process communication between MCP and the application's GUI.

This file will be edited when a new application is added to the environment.

2.2 The Graphical User Interface

MIDAS requires a consistent GUI, in the sense that a single GUI program must suffice for all applications. Therefore the project has developed a type of GUI which is table-driven, the tables being constructed from ASCII initialisation files. By providing an initialisation file, which controls not only the appearance and behaviour of the GUI, but also specifies templates for the commands which are sent by the GUI to the application, any relevant application may be introduced into the environment.

The table-driven approach is also extended to command decoders in MIDAS, so that the behaviour and functionality of these may easily be changed. These command decoders have the capability of recognising incorrect user input, and can issue error and warning messages via the GUI. Warning and error messages are provided by ASCII initialisation files and stored as tables, so they may be easily modified and extended.

A feature of the MIDAS GUI is the use of persistent menus. Many commercial GUIs make use of ephemeral or non-persistent menus, that is, the user may display a set of pull-down and pull-right menus, but as soon as an item is selected all the menus disappear.

Some features provided by the MIDAS GUI are:

1. menus, including radio buttons
2. dialogue windows for entry of values etc.
3. text windows for displaying messages, warnings etc.
4. sequencing of GUI objects.

If the application is provided with a suitable command decoder, then the user may use the GUI to enter parametric expressions instead of numerical data.

The MIDAS Environment was developed on UNIX systems, and makes use of X Windows for its graphics. However, all calls to the X Windows libraries are in a small subset of the program source files, at the lowest level. The Environment need not be restricted to UNIX; recently the GUI itself has been ported to Windows 95 and NT, using the Microsoft Win32 Application Interface.

3. Data Modelling and STEP

3.1 Data Management with MIDAS

The international standard for data exchange, ISO 10303, known as STEP [2],[4] is being used as an integration tool between CAD applications. At the start of MIDAS the initial release of the standard did not include finite element analysis but during the project's lifetime the FE became stable and is now a draft international standard [5]. However, electromagnetic analysis is not covered by the this STEP standard although the existing models and methodology can be used and extended into this area.

The overall goal of MIDAS was to design and implement an open client/server software tools environment. The tools were to be integrated in the environment within a framework that included the STEP compliant data management system. STEP was a very important component of the project as it was to provide an internationally recognised standard for data management. STEP data modelling methods and the SDAI data base interface were used throughout. As a result of this, the project developed a STEP data model including elements that will meet most of the needs of the electromagnetics community [3]. The MIDAS project has taken the current STEP standard and the STEP methodology and used them to extended STEP to provide a common data management system for MIDAS.

3.2 The Database

The prototype database used by the MIDAS environment was a system called DEVA. This was developed at Rutherford Appleton Laboratory and provides a database management system, I/O control and a data access interface. Within MIDAS, DEVA was enhanced to run as a client/server database so as to provide the foundation of the MIDAS environment. In this mode a server database is initiated on any accessible machine and then any application, acting as a client, has access to the data. Applications can then perform data management actions. For example: place data, access data, save data-sets and rollback.

When application processes are closed the server database remains active until explicitly closed. In parallel to this development an embedded version of the data base was continued as it was thought that the overheads of using a client/server data base might impact the performance of the applications.

A second important part of this development was the provision of an SDAI interface. This interface is defined in the STEP standard in both language binding and function. It therefore provides an important standard through which all applications can manage their data.

Like any other database, the DEVA database needs a schema which describes the data to be stored and accessed. DEVA is STEP compatible because it uses as a starting point the entity definitions of the Application Protocols and Integrated Resource Models of STEP. MIDAS has only added to and modified the Express descriptions to includeelectromagnetics applications.

Through the STEP tools associated with DEVA, in particular the Express compiler, a keyword definition file is generated using the MIDAS Data Model. This forms the basis of the DEVA data dictionary which drives the data management system. A crucial element of this process is the MIDAS Data Model.

3.3 Status of STEP data models

The STEP standard is made up of a great many parts [4] only a few of which are relevant to the MIDAS project. Apart from the formal definitions of the Express data modelling language those of greatest importance are:

- Part 42: Geometric and topological representation
- Part 45: Materials
- Part 104: Finite Element Analysis(FEA)
- Part 209: Application Protocol

At the start of MIDAS each of these parts of the standard were in a different state. Some of the parts are about to be released as an international standard while others are only in draft. During the project each of these parts of the standard were considered and where possible used in the definition of the MIDAS Data Model.

3.4 Data Requirements and Modelling

It was important in the MIDAS project to establish exactly what data each application process would need to be stored in the common DEVA database. If all the applications are to access their data from the same database then they need to have a common definition of all data including the data that is common to more than one application. This was achieved through careful consultation with the application providers. The output of this process was a detailed data requirement which required mapping onto the STEP standard.

As a result the STEP models are to be viewed as a starting point for each specific implementation. The entities and relationships described in these models need targeting. When deriving the MIDAS data models every effort has been made to keep as close as possible to the available STEP models. However, there are restrictions in DEVA, as in almost any database, which do not allow the full Express language to be implemented without some modifications. So the formal Express STEP data models were refined to produce the MIDAS Data Model. All the important data types and entities of STEP were retained but their definition was specified and augmented by the MIDAS specific requiements.

4. Geometric and Variational Modelling

In the past geometric modellers have almost always been designed as general purpose tools with no specific application in mind. The MIDAS modeller is thus rather different in that it is mainly targeted for use in supporting Electromagnetic analysis applications with requirements to model both object and free-space. This means that more emphasis must be on the ability of geometric functions in the modeller to deal robustly with 'pathological cases' of touching curves and surfaces and so forth. Furthermore, the type of shapes often involved in E-M analysis dictates accuracy requirements which are unusually stringent. Such shapes are characterised by features with a wide range of dimensions, typically a small air gap in an 'ordinary sized' artefact. Thus the actual requirement is for a geometric modeller with a large 'dynamic range' rather than the ability to represent small dimensions. A geometric modeller for general mechanical engineering applications would be expected to robustly handle objects with a ratio of maximum to minimum size of features of approximately 10^5, for the MIDAS modeller the target must be 10^7. The need to respond to the challenges just described can be localised within the modeller software by the 'separation of geometry and topology' throughout the modeller data structures and processes [6].

The development of the MIDAS geometric modeller has focused on four sets of requirements:

1. Direct integration with electromagnetic FEA
2. High accuracy of geometric algorithms
3. Openness and STEP based data
4. Integration of advanced features such as variational geometry for optimization applications

4.1 Representation of Geometric Models

The MIDAS geometric modeller uses a hybrid representation scheme combining Constructive Solid Geometry (CSG) and Boundary Representation (B-rep), STEP Part 42 [4]. As far as integration with FEA applications is concerned, the most important part of the model is the B-rep which is used directly in mesh genention.

Within the MIDAS environment, a general parametric representation of geometry is used, to allow generality, and an innovative representation of topology, in terms of non-manifold models. In the context of geometric modelling, the topological information describes the adjacency relationships between the topological elements (such as vertices, edges, faces), i. e., how these elements are connected, for example: the ordered list of edges that generate a face, or the group of edges to which a vertex belongs, or the vertices of an edge, etc. The different topological elements comprising the MIDAS B-rep data structure are illustrated in Figure 2.

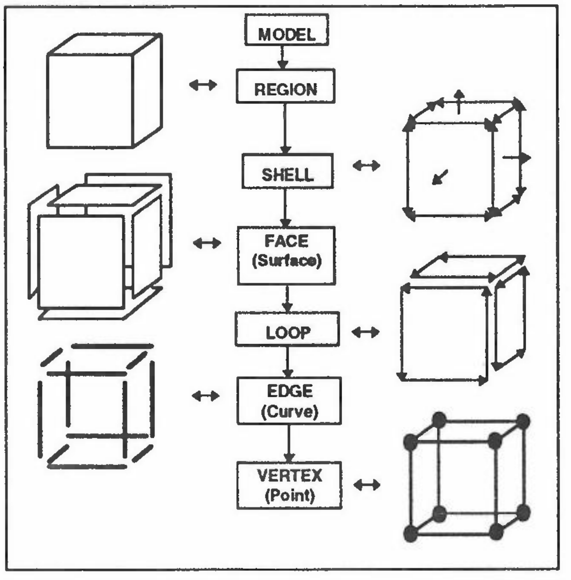

Figure 2: Top-down hierachial representations of some topological elements in the data structure

The primary goal is for the modeller to be able to represent non-manifold objects. Until recently most B-rep modellers could only represent manifold solids i.e. those satisfying the following constraints:

1. Two faces, two edges or one face and one edge of the solid intersect only at common edges or vertices.
2. Each of the edges is shared by exactly two faces, and each face belongs to only one finite region.
3. Faces around each vertex form a single circuit, i.e., a cycle such that each pair of consecutive faces in the cycle meet at an edge adjacent to the vertex.

The MIDAS modeller can represent any object that only satisfies the first of these three conditions. It should be noted that many other modellers cannot represent solids with a common face between two regions since they do not have the concept of a region in their topological structures. The MIDAS system can support the unified and simultaneous representation of wire-frame, surface and non-manifold solid models. Thus it is possible to represent a wire-frame and a solid with several regions sharing a common entity (vertex, edge or face) as illustrated in Figure 3. This facility allows more straightforward integration with the mesh geneator.

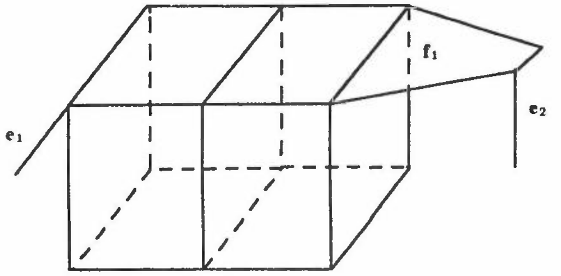

Figure 3: Unified representation of wire-frame, surface and non-manifold forms, with heteogeneous materials.

4.2 Manipulation of Geometric Models

As already stated all curves and surfaces implemented within the MIDAS modeller are or can be expressed in parametric form and this is exploited within the geometry module. The intersection of curves and surfaces is calculated using generic algorithms which *search in parameter space* to discover intersection points and curves. A more detailed description of such techniques may be found in reference [7]. A naive implementation of such algorithms can be slow, mainly due the large number of iterations involved. The approach adopted within the MIDAS modeller was to take particular care to preserve numerical accuracy in implementation and to attempt to speed up and improve the initial localisation of searches. These general methods were complemented by the development of *special case* code to deal with difficult conditions.

With respect to functions, they have been classified in several *layers*, going from general to more application oriented: data interfaces, algebra operators, geometric operators, topological operators, Boolean, sweeps, variational geometry operators, interfaces with mesh generation. The integration of Variational Geometry functionality should be emphasised, since the combination of these capabilities and the integration with FEA (finite element analysis) for electromagnetics opens the door for very advanced applications in the area of software for electromagnetic design.

4.3 Main Innovations of MIDAS Geometric Modeller

The MIDAS modeller provides a new non-manifold geometric modeller targeted at the particular needs of Electromagnetic applications. Its development also widens the area of application of geometric modellers. New applications will be related to the possibility for modelling both design abstractions (e.g. structures and components mixing solids and shells) and for modelling volumes with internal structure (e.g. composite materials).

The modeller development has also been characterised by the definition of new concepts, needed for high level operators, such as Booleans, and the integration of Variational Geometry. The risk inherent in such an ambitious development have been minimised through the organisation of the software development process and the structure of the modeller itself. Thus for example, the important functions for the MIDAS environment were developed independently of the Boolean functions.

5. The Mesh Generator

The fully automatic generation of high quality meshes plays a crucial role in FEM analysis, influencing both ease of use and accuracy of a software package. In this respect MIDAS proposes a very natural approach to the definition of mesh parameters followed by a very efficient meshing procedure from which the user is completely shielded. All data required by the surface and volume meshing algorithms is made available by the geometric modeller through a well-

defined set of interface routines providing FORTRAN access to the C++ structures of the modeller.
The first stage of the meshing process involves the definition of the subdivisions of the edges (possibly curved) surrounding the faces (the union of which then forms the different regions in the model). Biases can be assigned, resulting in the clustering of points towards one or both the endpoints of the edge (Figure 4).

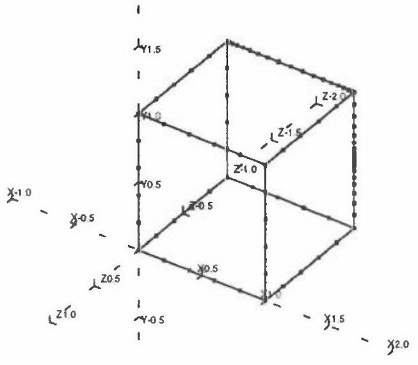

Figure 4: Regular and biased subdivisions on edges

The next step in the meshing process generates triangular meshes on all the different surfaces forming the regions. This stage is very similar to a normal two-dimensional Delaunay meshing algorithm, with the difference that the distortion between true space and the parameter space of the surface being meshed is taken into account, so that almost equilateral triangles are created in true space no matter how badly distorted they are in the local parameter space of the surface being triangulated. Also, a technique is used which automatically recognises if the surface discretisation does not properly represent the underlying surface, in which case surface points are automatically added.

An example of the effect of this algorithm can be seen in the slot on top of the cube of Figure 5. An alternative algorithm produces regular meshes on the surfaces by superimposing a regular grid of points on the surface and inserting them with a point by point Delaunay algorithm. The result of this process is a list of *required* edges and triangles, which should appear as part of the tetrahedral volume mesh.

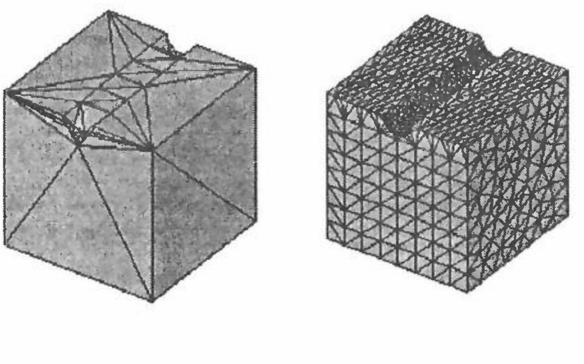

Figure 5: Regular, irregular and curvature-fitted surface meshes

The last step in the mesh generation process is the generation of the tetrahedral mesh itself. Three-dimensional simplexes were chosen as elements because they are very well suited for automatic mesh adaption and they easily support different shape function families (nodal, edge, facet).

The chosen algorithm is a multi-step one, which constructs tetrahedral meshes of good quality while preserving the user specified surface triangulation, which in fact guides the whole meshing process by imposing the mesh density. The first step involves a semi-constrained swap-based version of the Delaunay algorithm, which tries to produce a nearly-Delaunay mesh while preserving as many surface connections as possible, thus fairly well respecting the desired surface mesh. During this step, the incremental algorithm connects all the points generated during the surface mesh stage, thus filling the convex hull defined by the set of points. The swap-based technique chosen avoids most problems which are typical of deletion-insertion Delaunay algorithms The next step of the algorithm implements a modification of the surface restoration algorithm proposed in [8]. Once the prescribed surface has been reconstructed, with the possible addition of some internal points (Steiner points), it is easy to identify which elements belong to which region. Next, the mesh quality is improved by performing Max-min solid angle swaps, as suggested in [9], whereby high-order swaps have been avoided for the sake of efficiency of the algorithm.

The last step of the algorithm inserts internal nodes in the regions to provide a smooth transition of element sizes, according to the given mesh densities on the surfaces. During this stage Max-min solid angle swaps take place to ensure that only well shaped elements are formed. Table 1 shows the number of nodes and elements generated for the unit cube standard benchmark problem as a function of the number of boundary nodes (equal number of subdivisions on all boundary edges), CPU times obtained on a DEC Alpha 3000/300 are also reported, for the entire meshing process.

Edge sub-divisions	**4**	**8**	**12**	**16**	**18**
Nodes	13	740	2199	4945	6873
Elements	696	3990	12180	27843	38918
Time [sec]	6	10	25	60	67

Table 1 : Computer Statistics

6. Two-Dimensional Advanced Subset

The two-dimensional environment is fully integrated into MIDAS, and takes advantage of the same general features described in the previous sections, such as the database structure and the two-dimensional functions of the geometric modeling library. This was aimed to providing both more advanced functionality and greater simplicity of use to potential users of the two-dimensional subset, such as designers in small and medium enterprises (SMEs).

Because of the particular emphasis placed within the project on reliability and robustness, it has been decided to cast the error problem on a single element at a time, in order to avoid possible inaccuracies and topological complexities that could arise with multi-material domains in handling error problems defined on the *supporting regions* of nodes. Two error estimation procedures, that have proven the most reliable, have been selected and implemented in the MIDAS environment: the *Local Error Problem* and the *Local Field Error Problem*, evaluating errors in terms of potential and field, respectively. In both approaches, an *error problem*, posed in general and consistent terms, is set over each element, using as sources discontinuities on the element sides and residuals of differential equations. This problem is solved for the error, and then integrated over the element to obtain a normalized error indicator. For error in terms of potential, the error is represented over the element using the same shape functions used to represent the solution; for error problems in terms of field, the representation of the error variables is performed with edge elements, using *Witney* forms.

Error estimators have been developed for first and second order triangular meshes for plane and axisymmetric geometry. Error estimation algorithms are available for electrostatics, linear and nonlinear magnetostatics, including permanent magnets and surface currents, and linear steady state quasi-static magnetic problems [10],[11]. The estimators are then used to drive an adaptive meshing procedure, based on the subdivision of elements with errors higher with respect to a user-defined level, using an *h refinement* approach [10].

The adaptive procedures have been implemented in the two-dimensional MIDAS environment, and have proven very reliable and effective in the cases tested, these include a range of industrial design problems provided by MIDAS Consortium partners.

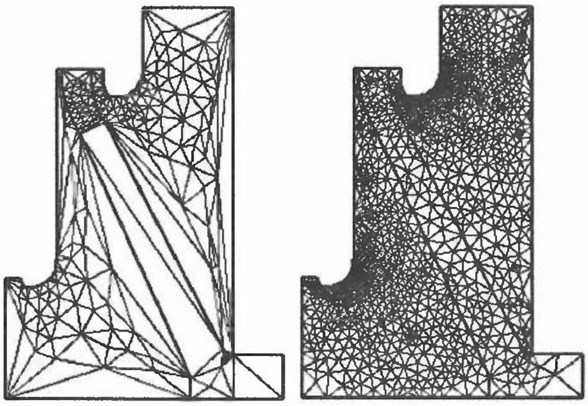

Figure 6: Initial and final mesh after adaption for the fields in a SF6 breaker. *Local Error Problem* error estimate, average error level 1%

The procedure has provided convergence, with six or less meshing iterations, using as starting point minimal meshes with nodes on the boundaries of the geometry only. An example of initial and final meshes for an industrial design test case is given in Figure 6.

7. Data Exchange & Applications

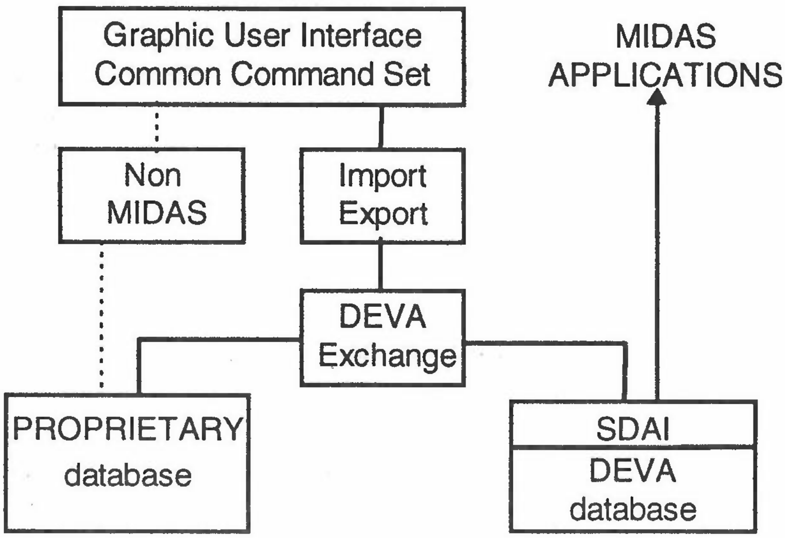

Figure 7: Using STEP to exchange data to and from a proprietary database

An important aspect of MIDAS is the rationalisation of data exchanging between proprietary environments. It is expected that Finite Element codes will evolve toward standardised data-structures, so provision of the protocols to interface proprietary environments in a robust way is vital for both small vendors of scientific software and large end-users. In the MIDAS system this has been achieved by implementing STEP-compliant data models that confer an openness by allowing data exchange at the database level, instead of at the pre-and-post-processing levels via neutral files. Once two databases adopt the same data-structure the presence of further interfacing sub-modules, as the exchange file depicted in Figure 7. will not be required anymore.

MIDAS will allow easy updates to the data-models as recommendations from the Standards committees emerge either by end users or the developers of the system. During the course of the project, the industrial partner involved moved from an existing system employing several interfacing routines which were heavily dependent on the code pre-and-post-processors evolution to the new MIDAS system which fulfilled the following three following main requirements:

1) Data models extended to non-electromagnetic analyses (structural & thermal)
2) Performing transient and multi-discipline analysis in amulti-tasking approach
3) Communication with the two-dimensional subset

Figure 8. shows an industrial example of a magnet system designed using components of the MIDAS

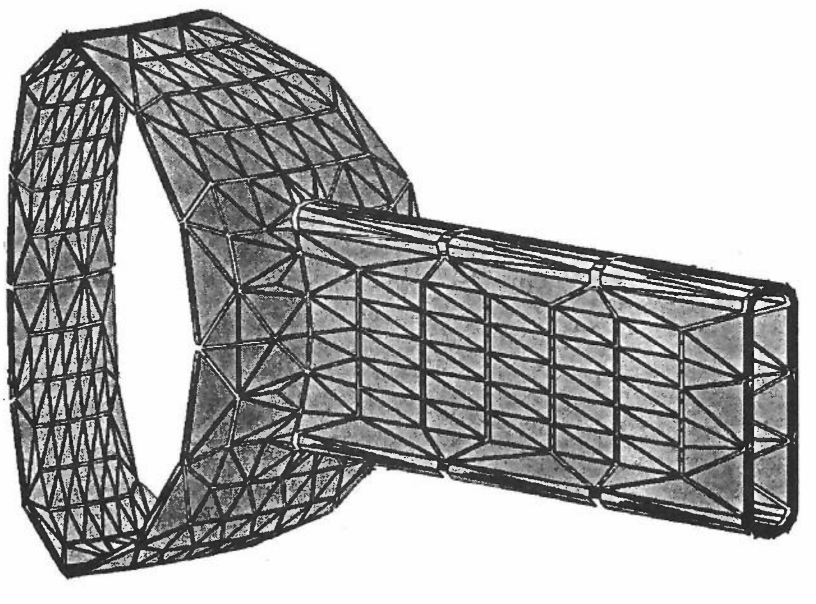

Figure 8 One sector of Tokomak case and access port [1]. MIDAS system used to perform eddy current and structural analysis of this structure

8. Conclusions

The MIDAS project successfully met its goals and the prototype software suite is currently in use by the industrial partners. A fully engineered version is currently under development.

9. References

[1] EC-ESPRIT Project 7294, contact for information, E Picco, Ansaldo Ricerche CPI, Corso F.M. Perone 118, Genova 16161, Italy.

[2] Jon Owen, *STEP An Introduction*, Information Geometers Ltd, 47 Stockers Avenue, Winchester, SO22 5LB,UK, 1993.

[3] D. Thomas and C Greenough, 'The MIDAS Data Model for Electromagnetic and Stress Analysis Integration', RAL.95.2 , March 1995.

[4] ISO 10303-11, 'Part 11: Description methods: EXPRESS Language reference manual'.

[5] ISO 10303-21, 'Part 21: Implementation Methods: Clear text encoding of the exchange structure'.

[6] ISO 10303-22, 'Part 22: Implementation Methods: Standard data access interface specification'.

[7] ISO 10303-41, 'Part 41: Integrated generic resources: Fundamentals of product description and support', IS Draft, 23 August 1994.

[8] ISO 10303-42, 'Part 42: Integrated generic resources: Geometric and topological representation', IS Draft, 29 August 1994.

[9] ISO 10303-43, 'Part 43: Integrated generic resources: Representation Structures', IS Draft, 15 August 1994.

[10] ISO/TC184/SC4/WG3 N258 , 'Integrated generic resources: Materials'.

[11] ISO TC184/SC4/WG3 N263x (P9), 'STEP Part 104, Integrated Application Resources: Finite Element Analysis', 9 September 1994.

[12] ISO TC184/SC4/WG3 N350 (P9), 'Part 209, Composite and Metallic Structural Analysis and Related Design', 17 October 1994.

[13] I.C. Braid, 'Notes on a Geometric Modeller', University of Cambridge, Computer Laboratory CAD Group Document 101 (1980)

[14] Hoschek & Lasser, Fundamentals of Computer-Aided Geometric

[15] *Design*, A.K.Peters, Wellesley, MA, USA, (1993)

[16] P.L. George, F. Hecht, E.Saltel, Automatic mesh generator with specified boundary', Comp. Meth. in Appl. Mech. and Eng., vol. 92, pp 269-288, 1991

[17] L. Kettunen, K. Forsman, 'Tetrahedral mesh generation in convex primitives' IJNME vol. 38, pp 99-117, 1995

[18] P. Alotto, P. Fernandes, P. Girdinio, A. Manella, P. Molfino, G. Molinari and M. Nervi: 'Error Estimate and Adaptive Meshing in Finite Elements solutions of Electric and Magnetic Problems', Proc. of Latsis Symp., Zurich (Switzerland), Sept. 19-21, 1995, pp. 1-20

[19] P. Alotto, P. Fernandes, P. Girdinio and M. Nervi: 'Mesh adaptation in Finite Element analysis of 2D steady state time harmonic eddy current problems', Conf. Record of COMPUMAG '95, Berlin (Germany) July 10-13, 1995, paper PC2-9, (to appear in *IEEE Trans. Magn.*)