Transactions on Information and Communications Technologies vol 2, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

An expert system application combining disparate software to form an integrated pressure vessel design system

I.J.S. Attwater, J. Anderson & G.E. Findlay
Engineering Design Research Centre, Glasgow, UK and University of Paisley, Paisley, UK

ABSTRACT

Over the past decade or so, the digital computer has enjoyed an ever increasing role in the field of engineering design.

Software packages developed for the designer are now abundant and have succeeded in automating certain aspects of the design process. They range from draughting packages which are run on personal computers to finite element analysis suites which require more powerful workstations to be run realistically.

These systems, however, do little to support conceptual design. They have been developed in isolation and hence seldom provide facilities to allow outside communication, and even then at a low level.

This paper describes a knowledge based system for the design of pressure vessels. In particular, it reports on the incorporation of, additionally, a finite element analysis package, which is automatically driven by an expert system shell. The expert system provides communication between the "islands of automation" which exist as disparate design packages and in doing so creates a framework for concurrent design of pressure vessels.

1. INTRODUCTION

The digital computer has had a major impact on automating the design process. The majority of their use has been focused on supporting draughting systems. These systems have been developed in isolation of each other and consequently have provided "islands of automation". Indeed in modern engineering design, CAD has been used almost exclusively in the phases of detail and embodiment design. Resulting intercommunication between these packages is not "built-in" and specialist interface programs are required to transmit information between them.

Expert system shells provide a vehicle for the construction of large scale design systems which can fully utilise and communicate existing software, thus providing a platform for the development of design automation.[1]

Pressure vessel design codes, such as BS 5500[2], have provided for some time rules and formulae for the design and construction of pressure vessels. Little or no support for conceptual design is provided in such codes and it is generally considered that the design methods embodied are sequential in nature. It is our considered opinion that in order to face a healthy future, the process industries must move to a concurrent state of design. The IMechE, enforcing this view, have said "...the process industries must adopt radical changes in their management of information technology if they are to stay in business."[3]

Central to a concurrent design environment is a global information model whereby the different design stages can communicate effectively. Expert system shells can provide a framework where data exchange and modelling standards can be integrated and, in so doing, instigate a common communication protocol used by the various stages in the design process.

2. REVIEW OF CURRENT COMPUTER APPLICATIONS IN DESIGN

2.1 GENERIC COMPUTATIONAL TOOLS

These tools generally come in the form of spreadsheets or maths libraries. Much of their application comes from domain knowledge which can be interpreted in an algorithmic form. This approach allows the tedium of iteration, which is usually associated with such design problems, to be avoided. They do not support any logical decision making but can be effectively executed repeatedly to obtain an optimised design, such as the required local thickness adjacent to a vessel nozzle.

Their use requires a high degree of understanding in the application domain but can also provide a standard answer form to the problem, advantageous in a consulting environment.

One of the authors has previously been involved in preparing such an automated system to specify the power requirement of a pump within a ship fin stabiliser assembly.

2.2 NUMERICAL ANALYSIS TECHNIQUES

Traditional analytical solutions to represent the response of components, such as pressure vessels, under certain conditions, can only be realistically achieved by the use of numerical analysis. This is due in part to the solutions either having a complicated mathematical structure or to the generation of a large number of algebraic equations.

The computer lends itself to this type of application readily and can obtain results in relatively short time intervals. Tooth and Nash[4] used this approach to provide a solution to a local load application on a cylinder. They used a Fourier series to represent the load and were able easily to increase the number of terms in the series.

2.3 THE FINITE ELEMENT METHOD

This method of analysis is gaining acceptance and provides a very powerful tool whereby the behaviour of complicated arrangements, such as complicated cylinder-to-cylinder intersections, can accurately be described.

Its use requires very powerful processors, for realistic applications, and tends to occupy very large amounts of media space. With the drop in real price of computer hardware, processing power is making it more accessible.

A high degree of skill is required to use these systems safely and quality standards are now just coming in to force. One criticism is that the new generation of finite element suites, with very powerful automatic features, require less skill for their use and therefore rigorous validation of the competence of practitioners is necessary.

2.4 PRESSURE VESSEL DESIGN AUTOMATION

Most pressure vessel design work in this country is carried out in accordance with BS 5500. Methods and formulae constitute current best practices and are based on research and experience.

Considerable domain understanding is required to use them and even then the solution path is often one of time consuming iteration. Here again the computer is very amenable to this type of application and high gains in productivity have been realised in their use.

Within these systems there is little support for conceptual design and a typical example would be that after having selected a suitable geometry, the required component wall thickness is computed.

A commercially-available example of this type of system is PVE5[5] for pressure vessels.

2.5 COMPUTER AIDED DESIGN SYSTEMS

Systems such as AutoCad[6] are widely used; their cost of ownership is usually a direct function of the level of 3D capability of the system. This has meant that these systems have been draughting, rather than design, systems.

The productivity impact of these systems takes place at four levels, as follows.

i. Improved accuracy, control and ease of modification through the flexibility and precision of these systems.
ii. Creation of standard libraries of graphical components (e.g., nuts, bolts and flanges) that can be scaled and added to drawings.
iii. Output of data downstream to manufacturing activities (e.g., profiles for numerically controlled machining).

iv. Creation of programs within the CAD systems allowing design knowledge and procedures to be committed to the system so that elements of the design process can be fully automated.

The majority of users usually achieve the first two levels; indeed failure to do so would make their investment worthless. It is only at level (iv) that these systems begin to impact on the design process, but this impact has been limited by the capabilities of the CAD system programming languages, the best of which have no better capabilities and facilities than a modern basic interpreter.[7]

3. PRESSURE VESSEL DESIGN PROCESS

A pressure vessel may be defined as a shell of revolution, which forms an envelope around some fluid, capable of withstanding internal or external pressure.

The basic criterion to be satisfied, in the design of a pressure vessel, is an assessment on whether the vessel will fail under the operating loads that will be applied over it's design life. The interpretation of "fail" in this case may involve bursting of the vessel shell or the attainment of a peak stress intensity. Design, applied to pressure vessel technology, is the assessment of the stresses and strains imposed by the applied loads and quantifying their significance with respect to the adequacy of the vessel to perform its intended function.

Materials selected will depend on the operating temperatures, pressures and the corrosive properties of the process fluid.

The aim of the reported design system, which will be described, is to provide a system that can be used to support the conceptual design of pressure vessels from the earliest stages within the design process.

Central to this system is an expert system shell which makes decisions on the basis of criteria such as design for manufacture, directs advanced stress analysis of vessel components using finite element methods, provides a concurrent communication environment and provides output in the form of a solid model.

4. ARCHITECTURE OF THE DESIGN SYSTEM

4.1 OVERVIEW OF THE SYSTEM

The system was built from the following major design support components:

i AutoSolid (Solid modeller and CAD system)
ii Nexpert Object (Hybrid expert system shell)
iii PVE5 (BS5500 and ASME VIII div 1 design code software)
iv Ingres (Relational database)
v Patran (Finite element system)
vi RDST (Step standard software)
vii EX (Step standard software)

The system was developed and runs under BSD4.3 UNIX on an APOLLO DN3500 workstation.

4.2 EXPERT SYSTEM SHELL

The central core of the design system is the expert system shell Nexpert Object[8]. It is a rule based and object orientated system which is capable of representing knowledge in a natural and powerful way.

The expert system provides a medium for representing, and conserving, design knowledge, and as a vehicle for the investigation of the science of design[9]. The expert system technology has emerged from research into Artificial Intelligence (AI).

The knowledge base is built of rules and objects. The rules form a network of conditions that can be traversed in the backwards or forwards direction to satisfy the design constraints. The object oriented data representation provides a very powerful way for modelling product design data.

Figure 1 shows the typical object oriented representation of a forged pressure vessel nozzle. The object is an instance of the class *forged_nozzle*. This class has been derived from the base class *Nozzle*. The object *Nozzle_1* will inherit the property slots of its parent classes. Thus the object has data slots which describe the object, and methods are attached to the object to manipulate that data, such as where to source it's value.

The shell, written in the "C" programming language, is provided as a development system and a set of libraries. This allows it to be used as an intelligent subsystem within an application, or as a means to provide the main system with the ability to communicate with other pieces of software. An example of the latter situation within the design system is where Patran, RDST and EX run as communicating subsystems from within the expert system shell.

Nexpert has a "callable interface" which allows functions that are written in the "C" programming language to be installed within Nexpert allowing program control to flow between nexpert and the installed functions.

4.3 COMMUNICATIONS BETWEEN THE SOFTWARE PACKAGES

The expert system communicates with the disparate design support systems through UNIX PIPES, which are interprocess communication channels. Pipes allow transfer of data between processes in a first-in-first-out manner and they also allow synchronisation of process execution. When the expert system is ready to direct a finite element analysis, for example, it creates a second process. To do this a "fork" system call is made which is the only way that a new process can be created in the UNIX operating system. The process that invokes the fork system call (the expert system) is called the parent process and the newly created process is called the child process. This second process is "connected" to it's parent via the UNIX pipes. Patran is executed in the child process and the standard input and output is duplicated allowing the input and output from Nexpert to be transferred through the pipes. In this way a two way

Transactions on Information and Communications Technologies vol 2, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

interactive dialogue exists in the same way as a designer would communicate with Patran if run independently.

Patran, for example, is run by the expert system as a separate process, and has had no modifications made to it's code to allow it to be used with the system. The interface system which controls the data flow through the UNIX pipes is written in the "C" and "C++" programming languages and are installed within the Nexpert "callable interface".

5. SYSTEM CONSTRUCTION AND DEVELOPMENT

5.1 CONSTRUCTION OF THE SYSTEM

Construction of this type of design system is a three phase operation that has the following sequence:

i. Indentify both the salient activities which govern the design process and the software to support these activities with regard to functionality and integration capabilities. Ascertain that the chosen software will integrate with the expert system; this will guard against "low-level" problems at a later stage.

ii. Integrate the expert system with the design software to provide a platform based on limited and simple heuristics. Test this initial system and ensure that it can achieve the design goals appropriate to this level of development.

iii. Provide a mechanism by which expert domain knowledge can be incorporated into the system and fully test the system against experts in the domain.

5.2 DEVELOPMENT OF THE SYSTEM

The current development of the system has concentrated on three aspects, the latter two providing a framework for concurrent design. Firstly, knowledge bases for the design for the attachment of nozzles to the pressure vessel shell and the application of local loads to these nozzles have been created along with the necessary interface code to successfully integrate with the existing system. Secondly, the major finite element software, Patran, has been integrated into the expert system. This integration, again seamless, allows communication between Nexpert and Patran through UNIX pipes. Finally, the emerging data modelling standards Step and Express have been implemented into the system, therefore creating a machine independent format for pressure vessel data exchange.

5.3 KNOWLEDGE BASE CREATION

As previously indicated, Nexpert supports object oriented representation of knowledge. The nozzle, local loads, "patran" and associated knowledge bases were created using this feature.

Like many other expert system applications it is necessary to ensure the design system is not over generalised and not reliant on a set of circumstances. A recognised method of avoiding this situation is to use dynamic objects. These objects inherit the same properties as their static counterparts but are created at run time. An obvious advantage of this is that the author of an expert system application is not required to "hard-code" the number of objects to be instantiated within a class.

Dynamic objects are referenced within the knowledge base by their "root" name, the common name, and their particular instantiation. For example, the base class *Nozzle*, with a property such as diameter, has a member object *nozzle_1*. The *nozzle_* part is the object root and the number following is the actual instance. This method is referred to as *interpretation* and *nozzle_1* will have it's own diameter property inherited from *Nozzle*.

PVE5, the design code software, supports a number of different types of pressure vessel nozzles. In accordance with point (ii) in section 5.1, work was concentrated on forged nozzles and in particular the Heavy Reinforced Neck type.

The resulting knowledge bases were prepared using a text editor and were integrated into the context framework. The necessary communication interface code to allow communication with PVE5 was written in "C++". Use was made of the dynamic structure of the objects by simply passing, from Nexpert, a temporary string containing the object root. "C++" string classes were used to build the full name of the object by concatenating the root with a loop counter. Access to the object's properties was made via Nexpert's "callable interface", and the UNIX program "awk" was used to parse the required information from PVE5's output.

5.4 FINITE ELEMENT INTEGRATION

As a widely used Computer Aided Engineering software tool, Patran[10] provides a vehicle for the automatic generation of finite element models. It enables the definition of the geometry to be made in parametric cubic space, allowing the "meshing" of the geometric model by choosing from a suite of available element types.

It was decided to implement within the design system the ability to direct a finite element analysis of the nozzle-to-pressure vessel shell intersection. The reason for this was twofold; initially, the authors have experience in modelling nozzle-to-shell intersections and it is accepted that BS5500 has limitations with respect to higher diameter ratios of nozzle-to-cylinder intersections. The finite element analysis will also be free from the approximations inherent in BS5500, to allow an analytical solution to be found.

The expert system has full control over Patran, with a full two way communication emulating a designer interaction with Patran. Patran is delivered with it's own programming environment Patran Command Language (PCL). PCL is a programming language designed to fit around the user interface of Patran and is similar in structure to the "C" programming language. Its main advantage is that it enables actual modelling commands to be embedded within the program structure, and thereby allowing the use of all the familiar programming techniques, such as loops, to manipulate the modelling facilities.

The reported design system uses the PCL feature to build an intersection model of the vessel shell and nozzle. The expert system provides to Patran, via PCL, information such as the shell thicknesses and diameters. The PCL program then automatically builds the model and submits the job for numerical analysis.

This method of communicating the modelling information to Patran avoids having to convert the data into a modelling standard such as IGES and is very light on available processing availability.

5.5 DEVELOPMENT OF MODELLING STANDARDS

Computer Aided Design system data exchange is based almost entirely on the exchange of geometric data and the bulk of that activity to date has been the transference of 2D draughting information. A large amount of effort, expended in building design systems of this type, is on the design of internal data models used by the particular system, and the transformation of that data so that it can be passed to another part of the system, or to other separate systems.

A very large part of development overhead associated with the design and implementation of internal product model representation in design and manufacturing systems could be avoided if product models were based on standard forms of representation and data exchange.

The ISO standard STandard for Exchange of Product data (STEP)[11] is beginning to provide a solid layer of enabling technology and standardisation on which to develop design systems. The STEP standard has "Object Oriented" features which allow near seamless integration with the expert system described in this paper. The design system uses STEP software developed at Rutherford Appleton Laboratories in the UK. The STEP software is divided into two parts namely Ex [12] and RDST [13].

Ex is an Express compiler. Express is the name of the formal information modelling language used to specify the information requirements of other parts of the STEP standard. In Express, entities are defined in terms of attributes; the traits or characteristics considered important for use and understanding. These attributes have a representation which might be a simple data type, such as integers, or another entity type. For example, a geometric point might be defined in terms of three real numbers representing their coordinates in 3D space.

The design system uses a structure describing the exchange of finite element entities and conforms with Express syntax. This Express definition file is passed through the Ex compiler which checks for correct Express syntax. The compiler's output is in the form of a Keyword Definition File (KDF) which describes the exchange structure by which data will be transferred. This operation is only required once as it does not describe the actual data but the structure in which future data will be exchanged.

The actual data which is to be communicated is written in a form which mirrors the exchange structure, the file thus created is referred to as the physical file. The software RDST (ReaD STep) is used by the design system to simultaneously read the KDF file and the physical file. This combination of

Transactions on Information and Communications Technologies vol 2, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

structures constitutes a virtual database and RDST provides a library of functions which are used to retrieve desired information.

The RDST functions are written in Fortran and the expert system communicates through a layer of "C" and "C++" interface code which was specially written. In the case of exchanging finite element data the expert system communicates with a Patran database through a library of Fortran functions called the DB-Access routines. Patran is delivered with these functions in a compiled library, and again more interface code was written, in "C" and "C++", to allow communication between the Patran database and Nexpert.

The above arrangement allows a concurrent communication environment to exist. The expert system can either create a physical file from an existing patran database and therefore exchange this data to another part of the design system (or application) or use the RDST software to read in a physical file and create a finite element model from the exchanged data. In both cases the data exchange conforms with the STEP standard.

6. CONCLUSIONS AND DISCUSSION

Figure 2 shows a wire frame output of the designed pressure vessel. The vessel is 10 metres high with a diameter of 2 metres. The ends are closed by two hemispherical heads and there are four forged nozzles in the vessel shell. The vessel shell has been thickened, by the system, in places to accommodate the various nozzles.

Figure 3 shows part of the finite element model, produced by the system, of a nozzle-to-vessel shell arrangement. The model is made up of Hexahedral 8-noded solid elements with each node having three translational degrees of freedom.

This project has gone through two levels of development and has shown that it is possible to build sophisticated large scale design systems without resorting to enormous amounts of low level code.

It is becoming apparent that CAD system data exchange is an important issue. Many CAD system vendors are now including the ability to output CAD information using the STEP standard. It is expected that this trend will continue and this project has shown that expert systems can be developed to use the features that this standard affords. Concurrent Engineering is a field which is receiving much attention. STEP has been used in this research and expert systems, in using this standard, can provide a vehicle for a concurrent communication environment.

It appears to the authors that very little research has been directed at creating an expert system for finite element analysis. This project addresses this issue and has shown that such a system is feasible. There is an important role for the STEP standard in such a research programme, enabling the avoidance of the recognised problem of enormous amounts of model information which previously had to be exchanged.

7. ACKNOWLEDGEMENTS

The authors wish to acknowledge the support and work done by John Lawson.

The project described was funded by SERC grant, SERC-GR/E/85-614 and by a University of Paisley research assistantship.

The authors would also like to acknowledge the support given by Whessoe Technical and Computing Systems Limited and Babcock Energy plc.

Finally they would like to thank and acknowledge all of the help given by academics in various institutions and by industrialists.

REFERENCES

1. Lawson, J., Anderson, J., 'Knowledge Based Systems for Pressure Vessel Design' *Int. J. Pres. Ves. & Piping,* Vol.53, pp. 23-38, 1993.
2. British Standard BS5500:1991, *Specification for Unfired Fusion Welded Pressure Vessels* British Standards Institution, London, 1991.
3. Computer Applications Committee of the Institution of Mechanical Engineers. *A Five-year Strategy Programme for the Process Industries,* 1988
4. Tooth, A. S., Nash, D., 'The Use of a Microcomputer in the Design of Cylindrical Pressure Vessels.' *Int. J. Pres. Ves. & Piping,*Vol 24, pp. 1-12, 1988.
5. *PVE5.* (Pressure vessel design code software to BS5500 and ASME VIII division 1), Whessoe Technical and Computing systems Ltd, UK.
6. *AutoCad.* (CAD system software), Autodesk Ltd, UK.
7. Anderson, J., 'Combining Disparate Packages to Form Integrated Design Systems.' *Keynote Paper, PEDAC '92,* University of Alexandria, Egypt, 1992.
8. *Nexpert Object.* (Expert system shell), Neuron Data, Palo Alto, California, USA.
9 Rychener, M.D.(Ed). Research in Expert Systems in Engineering Design. *Expert Systems for Engineering Design.*, Academic Press, London, pp. 1-33, 1988.
10. *Patran Version 2.5.* (Finite element software), PDA Engineering Ltd, Costa Mensa, California, USA, 1990.
11. STEP, Standard for Exchange of Product Data. *International Standards Organisation, ISO 10303.*
12. *EX.* (STEP software), Rutherford Appleton Laboratories, UK
13. *RDST.* (STEP software), Rutherford Appleton Laboratories, UK.

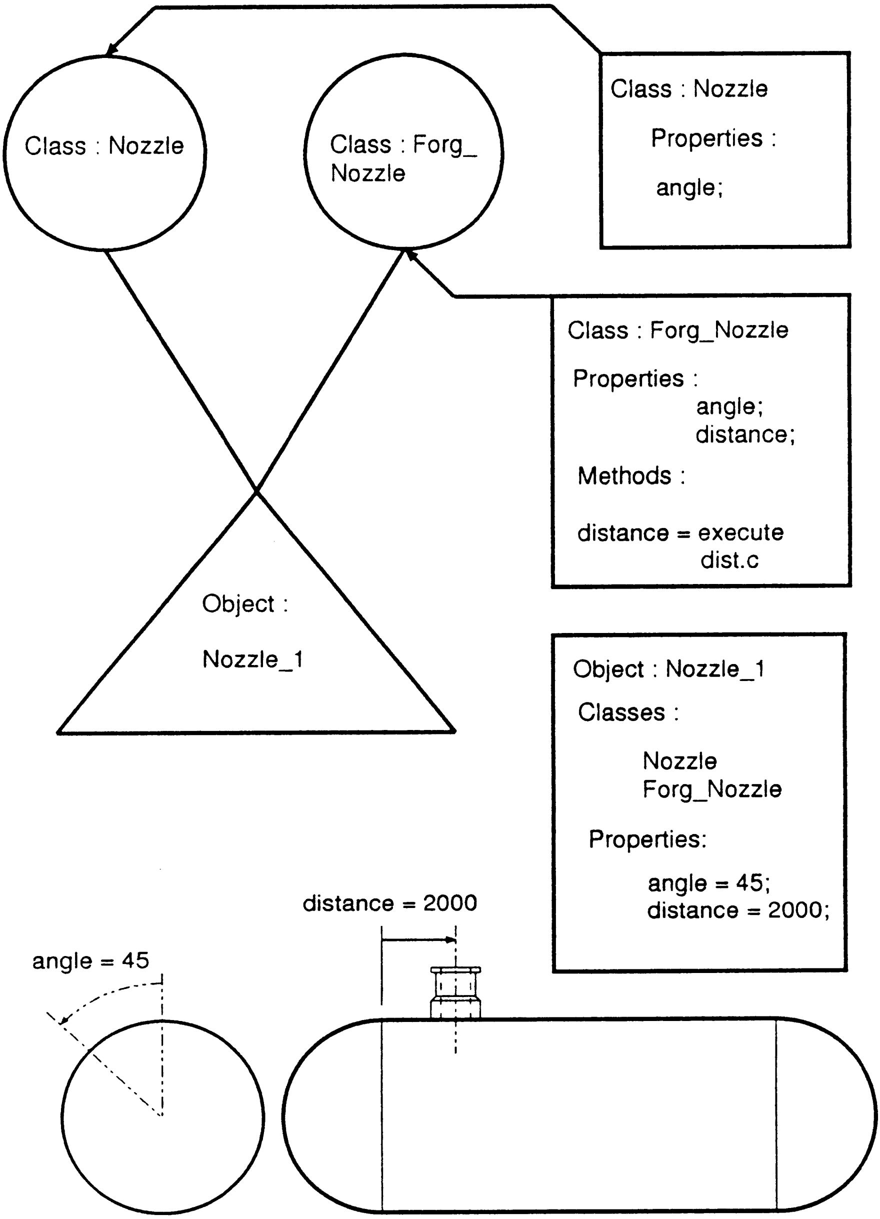

Figure 1: Example of Class hierarchy of a nozzle component.

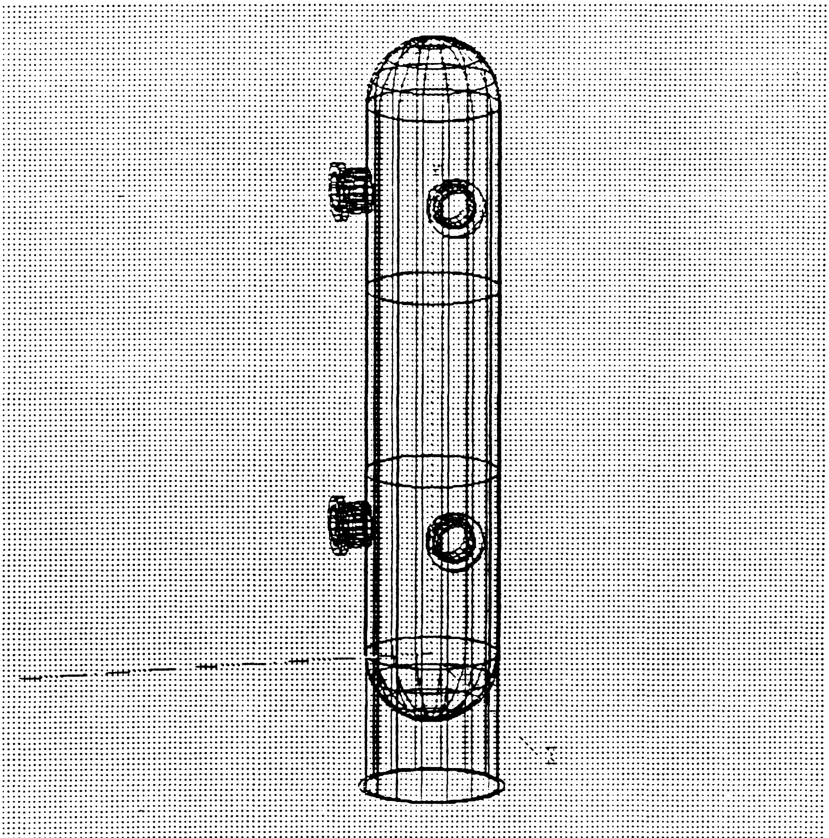

Figure 2: Example Output from the System.

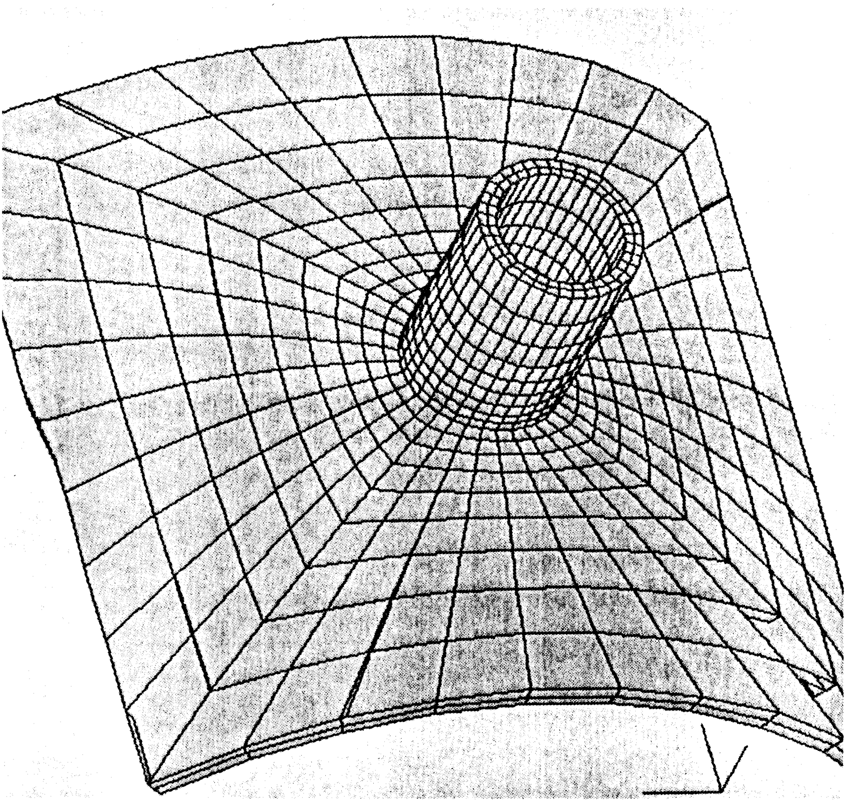

Figure 3: Finite Element Model of a Shell-to-Nozzle Intersection Created by the System