Integrating PATRAN with CSR Software for Large Ship Strength Assessment

Daron H. Libby
DHL Consulting
August 2011

ABSTRACT
The Common Structural Rules Software is an integrated software package covering the strength assessment of double hull oil tankers and bulk carriers using finite element analysis. The IACS Common Structural Rules (CSR) software draws upon the technical strengths of the American Bureau of Shipping of Houston, Texas and Lloyd’s Register of London, UK and will be used to evaluate new designs presented to either classification society.

Many of the world’s largest shipyards use MSC/PATRAN and NASTRAN for finite element modeling and analysis. A robust and complete user interface to PATRAN was required as an optional CSR Software preference.

This paper discusses the 12-month multi-national (US, UK, India, China, Korea, Japan) project to design, develop, and test the PATRAN-CSR interface, which comprises 22 PCL customized forms and some 25,000 lines of PCL code to support modeling verification and results evaluation of large bulk carrier and oil tanker finite element models.

Many PCL coding obstacles were confronted and overcome using innovative PCL programming solutions in order to meet the needs of the ship design analyst, such as:

- Accessing primary and ancillary databases within a single PATRAN session
- PATRAN Client Data utilities for time stamping data blocks
- Derivation of mean stress tensors
- Interactive editing of element attributes without using element properties
- Display of opposing force vectors at a single node
- Creation of color-coded polygons to represent face plate stress in bars
- Implementation of MSC/Nastran and MD/Nastran restart

Many of these PATRAN customization techniques through PCL programming could be applied to other engineering disciplines.

The CSR-PATRAN interface was released worldwide in July, 2011.
**Background**
The U.S. consumes about 20 million barrels of oil a day. For some perspective, China is second at just 8 million barrels a day. Supertanker capacity varies, but most carry about 2 million barrels. The largest tanker in the world carries 4 million per day. So, it takes about ten “normal-sized” supertankers to supply the U.S. with oil for a single day. As the demand for energy, and mined and manufactured products increases, the world's demand for more oil tankers, bulk carriers, and containerships only continues to grow. Hyundai Heavy Industries in South Korea, with nine large-scale dry docks and seven huge 'Goliath Cranes', is the world’s largest shipbuilding company and builds about 60 new ships a year.

**Introduction**
The proportions of Noah's Ark are explicitly stated in Genesis 6:15 as 300 x 50 x 30 cubits. The vessel is ten times as long as it is high, which means that the bending loads applied by waves are significant. The Ark had similar proportions to a modern ship, and ships are not supposed to break in half!

To avoid making a ship that is too weak in the middle, there are Rules.

Over the past century, a ship owner would work with an independent Classing Society on the design of a new ship. This procedure classed a vessel for safety from the time the keel was laid, through sea trials, to the time when the ship became operational and the owner could procure insurance. There are less than a dozen major Classification Societies in the world and all had their own interpretations of Rules for the design of large ships, namely oil tankers, bulk carriers, and container ships.
Over the past 20 years, Class Societies became more competitive to be able to quickly produce a design assessment for a ship owner or shipbuilder. Also, depending on a particular ship type and how the Rules were interpreted, an owner found the total steel requirement could vary from one society to another. This competition among Class Societies influenced them to reduce the number of load cases considered and to interpret the rules in a way to minimize steel requirements.

In the early 21st century, the International Association of Class Societies’ (IACS) developed Common Structural Rules (CSR) that applies to all bulk carriers equal to or above 90 meters in length and to all double hull oil tankers (DHOT) equal to or above 150 meters in length. For a sense of size, a modern ultra-large crude carrier (ULCC) can be 1,300 feet (400 m) long and have a capacity of 500,000 DWT (dead weight tonnage).

**What is CSR?**

Many of the rules of IACS Class Societies are largely prescriptive, which first started with experience-based maintenance records and evolved to a combination of empirically and theoretically derived rules and requirements.

The new Common Structural Rules are a welcome initiative for the industry, as they have been developed using a consistent engineering approach that applies advanced analytical and numerical methods based on engineering first principles to establish the strength requirements. In short, CSR may be summarized to yield the following characteristics:

- Rules covering structural requirements for bulk carriers and tankers.
- A Rule set utilizing Finite Element Analysis for more extensive direct strength calculations.
- Vessels built to CSR shall have overall safety of the hull structure equivalent to or better than that currently achieved by any current rules by any Class society.

The reason for introducing CSR may be summarized as follows:

- To obtain improved control over the minimum safety level during the operational phase of a ship’s life.
- To eliminate competition between Class Societies with respect to structural requirements and design standards.
• To employ the combined experience and resources of all IACS societies to develop a single standard or set of Rules.
• To ensure that a vessel meeting this new standard will be recognized by the industry as being at least as safe and robust as would have been required by any of the existing Rules used by a classing society and to eliminate competition on ship scantlings.

New PATRAN Interface with unique support for CSR
The new Rule requirements for bulk carriers and oil tankers introduce a radical shift towards more computerization of the rule formulations and structural assessment. Hence, good software support is critical for any classification society to provide timely and accurate support to ship owners, designers, and shipbuilders.

CSR Stage 2 & PATRAN Interface

CSR Development Team
The American Bureau of Shipping (ABS) and Lloyd’s Register (LR) announced in April 2011 the release of CSR Software, a jointly-developed system to be used to assess bulk carriers and oil tankers designed to comply with IACS Common Structural Rules (CSR). The common software draws upon the technical strengths of ABS and LR and will be used to evaluate new designs presented to either society. It is the desire of ABS and LR that other societies join in this initiative.

Recognizing the strengths of their collaborative efforts and to demonstrate commitment for consistency, ABS and LR also announced in April 2011 the establishment of Common Structural Rules Software LLC – a joint venture company with offices in Houston and London. The entity maintains the newly released software and develops new common software which will assess vessels designed to comply with the harmonized CSR that will be submitted for industry review in 2012.
Where to Start?

Years prior to the April 2011 announcement, both Lloyd’s Register and ABS had legacy modeling and analysis software available to use in the new CSR product, and work began to merge the best of both companies software into a new analysis system. There had been an initial effort to provide graphics support for the CSR computational program using Open Inventor. But, due to the complexity of user requirements, it became apparent to ABS engineers that FEMAP could be modified to provide much more powerful graphics tools; a program that they were very familiar with.

When Lloyd’s Register engineers began to look more closely at the new CSR program in the fall of 2009, they quickly called for a similar PATRAN Interface, as they were primarily expert in the use of PATRAN and quite unfamiliar with FEMAP.

Time was of the essence as ABS was given the task to produce a PATRAN Interface for the new system by March 2010 … just three months away. The first knowledgeable PATRAN programmer to look at the job estimated over one man-year effort to develop the PCL - FORTRAN interface described in the CSR-PATRAN Interface product specification.

In late December, 2009, the author was called about an inquiry of a PATRAN Interface to an ABS product SafeHull which was developed in 2002. The new CSR modeling and analysis software would use much of the same workflow and the thought was that perhaps this legacy PATRAN Interface could be used with some modifications. But an archive search for PCL source at ABS headquarters in Houston had turned up empty.

The author flew to Houston in December for the initial fact-finding meeting. Indeed, the project required PATRAN to read and display, and in some cases edit, CSR files in 15 points in the workflow of modeling, analysis, and post-processing of computed results. The estimate of one man-year was probably correct. The PCL of 2002 looked like it would save a lot of time if it could be found. The author returned to his office in New Hampshire and after a two-day search on several old computers and CDs, it was an unlabeled 3.5” floppy that was found to hold 15 PCL files, each containing 1000 lines of PCL code. We were in business!

Most of the CSRS to FEMAP interface had already been developed by ABS engineers in Houston and the instructions to the author were, “Whatever FEMAP can do, PATRAN must do.”

For two months, the author and Michael Schmidt, who is an ABS contract employee and FORTRAN programmer, worked overtime to get the new PATRAN Interface ready in time for the March release. The hooks to PATRAN inside the CSRS program were developed in Chennai, India. One man-year of development was accomplished in two months and the PATRAN-CSRS Interface was ready for testing in March of 2010.

The March testing went quite well and resulted in new requirements for 30 fixes and interface enhancements, which were added in the Spring. This second phase was completed in time for the first User Acceptance Testing (UAT) in Shanghai during June of 2010. This UAT, attended by top shipyard and technical office analysts from London, Japan, South Korea, Houston and China led to new product specifications for 25 additional fixes and enhancements for the PATRAN Interface which were developed during the Fall in time for the second UAT in January, 2011. Another 40 requirements were identified. Due to the
“product enhancement creep”, a new project manager of the newly formed CSR Software LLC in London, came onboard. He tirelessly facilitated one or two conference calls a week between London, Chennai, India, New Hampshire, and Houston for nearly nine months and kept the project on time, on budget, and moving ahead. Software testers were brought in from Lloyd’s Register offices in London, Shanghai, Busan, South Korea, and Yokohama, Japan. The latest CSR Software Version 4 was released in July of 2011. What started with 15,000 lines of PCL in March of 2010 had grown to 22 customized PATRAN forms and nearly 30,000 lines of PCL in this latest version.

**CSRS Architecture**

The CSRS program is a C#/Net application and is separated into a Stage 1 application to specify design dimensions and estimated scantlings for prescriptive rule assessment and Stage 2 application for finite element based strength assessment. The CSR Stage 2 workflow leads the analyst from model import, through loading of 40 or more load cases, to NASTRAN analysis, and through yield, buckling, and fatigue assessment.

A typical CSR Stage 2 finite element model comprises the middle three-hold section of the ship (where the maximum bending moment occurs) and consists of stiffened plates (shell and beam elements) with the number of finite elements anywhere from 70,000 to upwards of 500,000. The model is subjected to at least 40 load cases and the new Harmonised CSR planned for 2012 considers over 100 load cases.
The GUI selections in the CSRS program result in:

a. sub-GUI menus and/or Visual Basic displays
b. launch batch files to spawn Fortran programs for numerical processing
c. spawn the graphics program to display images necessary for data evaluation for the step in the workflow

A graphics preference can be set to either Open Inventor, FEMAP, or PATRAN, and in CSRS version 4, PATRAN-ADVANCED.

Within the workflow, there are more than a dozen instances where PATRAN is called upon to read in data, edit and return data to CSRS, or display customized evaluation plots.

- View Model & Groups
- View/Edit Corrosion Data
- View Tanks
- View Tank Loads
- View Loads & Boundary Conditions
- View/Edit Buckling Panel Definitions
- View/Edit Panel Attributes
- View/Edit Yielding Element Attributes
- View Solution
- View Buckling Evaluation
- View Yielding Evaluation
- View Fatigue Evaluation
Launching PATRAN from with CSR Stage 2
When the CSRS user is ready to see something in PATRAN, the appropriate icon or GUI menu button is selected. A batch file is written and executed.

This batch file first calls any Fortran programs required to read CSRS data and prepares input data for PATRAN, writes a one-line PATRAN session file, and then launches the PATRAN program. When PATRAN comes up, it opens a new database, and automatically reads the session file of the form:

```
>>p3csr_view_tanks.input("project_path", "shipname", "shiptype", "flag")
```

which provides entry into the PCL Class to read in data and display a customized GUI within PATRAN.
Here is a typical customized PATRAN GUI for (1 of 22 in the Interface) for manipulating the CSR data in PATRAN.

When the user has completed the PATRAN task, the Cancel button is selected, PATRAN exits, and the user returns to the CSR Stage 2 program.
Customizing PATRAN to meet Requirements

Many PCL coding obstacles were confronted and overcome using innovative PCL programming solutions in order to meet the needs of the ship design assessor, such as:

- derivation of mean stress tensors
- display of opposing force vectors at a single node
- creation of color-coded polygons to represent face plate stress in bars
- implementation of MSC/Nastran and MD/Nastran restart
- interactive editing of element attributes without using element properties
- display of irregular-shaped panels
- accessing primary and ancillary databases within a single PATRAN session
- PATRAN Client Data utilities for time stamping data blocks

Many of these PATRAN customization techniques through PCL programming can be applied to other engineering disciplines.

1. Derivation of mean stress tensors

The results database (.xdb) created during the NASTRAN analysis writes top and bottom stresses for each plate element. For rule compliance assessment, the average of top and bottom stresses (or mean stress) for each plate element are used when considering yield.

To handle this, the following PCL calls create and display the mean stress vector for the current load case:

- **res_data_load_dbresult** - loads in Z1 results
- **res_data_dbres_list** - loads in Z2 results
- **res_data_list_average** - computes the vector of mean results
- **res_db_create_loadcase_c** - create the temporary loadcase for Mean_Results

Below, a plot of mean stress on the deck is shown in local element directions.
The beauty of this approach is that not only can the user display the mean value for each element, but also the Results Transformation from Local to Global coordinates can be easily turned on on/off with a flag in the call to `res_data_load_dbresult`.

2. Display of opposing force vectors at a single node

It is common for a single plate element to have pressure applied to both the top and bottom surface, for example, when two loaded tanks share a tank wall. In CSR Stage 2, all frame loads are converted into applied nodal forces. For load verification purposes, the user requests to see opposing force vectors at a single node. If two forces applied to the same node are read into PATRAN, they will always be automatically combined and the resultant force is displayed. It is impossible for PATRAN to carry and display two or more forces at a node for a single load case.

To get around this limitation, the FORTRAN program that writes CSR Stage 2 nodal forces into the PATRAN neutral file for input to PATRAN, writes the actual loaded node the first time it is encountered in the CSR Stage 2 frame load data. All other forces applied to the same node (due to an adjacent loaded tank) have the node ID offset to be a unique ID. The new phantom node definition is written to the PATRAN neutral file along with the force packet for that phantom node.

When the neutral file is read into PATRAN, multiple forces appear at a common spatial location, as shown in the following plots.
Above, frame loads (nodal forces) are depicted as yellow vectors. It can be seen that opposing forces along the centerline longitudinal bulkhead are displayed using a duplicate, phantom node technique.

In the 3D plot below, three nodal forces can be observed at a single point in space.
3. Creation of color-coded polygons to represent faceplate stress in bars

Ship structures commonly have faceplate stiffeners along the open edge of a steel panel. These faceplate stiffeners are welded orthogonal to the panel and are typically modeled as bar elements. The axial stress in the faceplate is computed with the bar element and the results are very accurate. However, it is post-processing that presents the challenge as the rod elements do not stand out in a large model.

FEMAP has the ability to plot axial stresses in rod elements as colored polygons. This kind of display was required in PATRAN, where the color of the polygon represented the stress level and the relative size of the polygon represented the magnitude of axial stress. PATRAN does not have such a tool.
To meet this color-coded polygon plot display challenge, PCL was written to create the plot following this procedure:

a. the axial stress for each rod element in the current group is retrieved from the results database (.xdb) using the PCL function res_data_load_dbresult.

b. the max and min result for the current group is extracted (res_utl_extract_elem_results)

c. the planar orientation of the current group is determined by inspection of nodal coordinates. Then the vector normal to this plane is determined. (db_get_nodes)

d. the element results are normalized to pscale, which is set by the user.

e. for each bar element in the current group, compute the cross product of bar axis and out-of-plane vector. Sweep a dummy quad from each bar element using the normalized stress value for sweep height. (res_utl_cross_results & fem_sweep_extrude_1)

f. use dot product to ensure all polygons sweep in the same direction. (res_util_dot_results)

g. store the axial result as a result for the dummy quad (db_add_s_elem_res_by_type)

h. display the fringe plot (res_display_fringe_post)
4. Implementation of MSC/Nastran and MD/Nastran restart

In 2003, the PATRAN interface was looking to use the NASTRAN random access results database (xdb) for all post-processing. The ABS analysis procedures all used the NASTRAN Output2 sequential text format to store the raw NASTRAN results. So, at that time, the batch files for NASTRAN solution ran the job twice when the PATRAN preference was enabled—the first job created the OP2 file (PARAM,POST,-1) and the second run changed the PARAM,POST, 0 card and ran the job again to create the xdb file.

In 2010, LR engineers questioned why the NASTRAN job was being run twice. Over the past seven years, the FEM models became larger and the extra execution time to run the same job twice was now a concern. The author, working with LR engineers, created a NASTRAN restart procedure to run the first job and save the restart files. Then the batch file changed the PARAM, POST, 0 card, performed a restart, and the xdb file was created in seconds.

However, further testing indicated the procedure only worked for MD/NASTRAN and failed with MSC/NASTRAN due to the fact that changing any card in the Bulk Data section of the input file (including just a PARAM card) caused MSC/NASTRAN to think the model had been changed and it ran the whole job all over again. After more study, the PATRAN Hotline was contacted. MSC support engineers located a rarely-used Bulk Data card they called the “slash command.”

The restart job was altered to include the following lines:

```
/, 1, 9999999
PARAM, POST, 0
```

As in the MSC/NASTRAN user manual\(^1\), the slash command is actually documented as a Delete card, used to remove entries on restart: / K1 K2, where this entry causes Bulk Data entries having sort sequence numbers K1 through K2 to be removed from the Bulk Data upon Restart. K2 may be specified as larger than the actual sequence number of the last entry. This is convenient when deleting entries to the end of the Bulk Data Section.

With this tip from the PATRAN Hotline, a test of a large double-hull oil tanker model showed the first run to solve the model and create the OP2 file took 45 minutes. The Restart ran in 45 seconds and the .xdb file was created.

5. Interactive editing of element attributes without using element properties

Part of the CSR Stage 2 model definition tool is to assign corrosion data to each element and scalar attributes to panels (used in the evaluation of panel buckling and yielding). The PATRAN requirement was to:

a. display these attributes assigned to elements and panels as color-coded plots
b. allow the user to edit these attributes and write this data back into the CSR Stage 2 system files

The problem arose out of the fact that PATRAN uses “element properties” to assign scalar values to the elements. In other words, all elements sharing a common property are specified as the elements assigned to that particular property type. However, the CSR

Stage 2 system specifies seven corrosion attributes for each plate and stiffener and, in the case of the bulk carrier ship type, over thirty attributes for each panel. To utilize the notion of PATRAN properties would require a huge looping program to inspect each attribute for each element or panel, determine if a property for that value had been created, and eventually group elements sharing a like attribute into property sets. Then once in PATRAN, if the user wanted to change an attribute for a single element, this again violates the PATRAN design where PATRAN prefers to change the property and that affects all elements in the property set.

This attribute editing capability had to be in the March, 2010 release which had a six week window for PCL development. In one week, PCL was developed to read in the attributes for each element, store the attributes in a huge array, 30 words for every element, store the attributes in PATRAN as discrete data fields, use dummy TEMPERATURE loads to represent element scalar values, and assign the fields (one field per attribute) as temperatures.
The plot was created and the user is given PATRAN selection tools to select elements, query current attribute values, and to even modify attributes to the elements selected. Upon Apply, the field for the current attribute is deleted, the temporary array is updated, and the field is recreated. Finally, the plot is updated to show the modified attributes.

Upon PATRAN exit, the user is asked for confirmation to modify the CSR Stage 2 data files and then the modification is completed using PCL text file commands.

6. Display of irregular-shaped buckling panels
The panel is a region of stiffened plates that is typically bounded by main longitudinal and transverse stiffeners. Special buckling and yielding evaluation rules apply to panels, which must be checked during the strength assessment process.

The CSR Stage 2 system has tools to automatically define most of the panels required for evaluation. The PATRAN interface needed the ability to display and edit these panels, which consist of anywhere from one to more than 30 finite elements in a panel.

Because over 90% of the panels in the ship structure are rectangular, the panels were represented in FEMAP as “special” quadrilateral elements. This same approach was used in the initial PATRAN interface. However, there are always some irregular-shaped panels that can have many more sides than four and a quadrilateral cannot be used to show the panel shape. Beyond that, the analysts wanted a button to control the display of the actual finite elements that make up the panel. This requirement mandated that the special quadrilateral approach be eliminated and replaced with a more elegant solution.

PATRAN has the ability to create a multi-sided trimmed surface whose outer shape exactly matches the envelope of a collection of finite elements (sgm_create_surface_from elems2). Once the surface is created, a second PCL command can be used to associate the elements with the panel (sgm_assoc_element_surface_v1).
Using this technique, the panel is accurately displayed, the elements that make up the panel can be toggled on/off, and new panels can be created in PATRAN and written out to the CSR Stage 2 system. The red multi-sided polygon is a single panel and the elements are toggled on/off.
7. Keeping the PATRAN process open - The Advanced User Interface (AUI Module)

In the initial design of the PATRAN Interface for CSR Stage 2, the same procedure was followed as had been implemented for use with FEMAP. Namely, each time the user requested a graphics display in PATRAN, the batch file was executed, the PATRAN process was started, a PATRAN database was opened, and data was read in. Even with models on the order of 200,000 elements, the time to start PATRAN and read in the NASTRAN model was perhaps five to ten seconds. When the user finished with that PATRAN session, the “Return to CSR” button was selected, PATRAN exited, and the user was back in the CSR Stage 2 program. When another PATRAN request was made, the whole process was repeated.

However, this became a problem well into the UAT testing, where users were checking out their PATRAN license on a remote server. First of all, the user might have to wait upwards of five minutes to obtain a PATRAN license, and then it took time for PATRAN to start up and read in the model. Users complained of having to start and stop PATRAN perhaps a dozen or more times in a typical CSR Stage 2 session and the current interface architecture was deemed unacceptable in a production environment.

The author first attempted to use a polling technique, in which the first instance of PATRAN would start a polling function, where every five seconds PATRAN would look to see if a “wakeup” file existed. When the user requested a PATRAN display, the batch file would create the “wakeup” file, and in seconds PATRAN would see it, would immediately read in the session file, and come to life. When the user hit the Return button, PATRAN would delete the “wakeup” file, call the polling function, and “go back to sleep.”

Although this worked, there was no way to “push” the PATRAN window to the front once it came to life. Also, if the user killed the PATRAN process using a Windows widget and not the GUI “Return” button, the “wakeup” file did not get deleted and the polling function was not started and things got hung up. This technique was deemed too unstable and so was not used.

To get around the problem, a shared data link system was developed between CSR Stage 2 and PATRAN for the expert PATRAN user who understands how the two programs are sharing data. The system was called the Advanced User Interface (AUI Module) and its main PATRAN PCL utility used was the Client Data functions.

Client Data functions are PCL calls to store and retrieve an arbitrary collection of integers, reals, logicals, and strings which are stored in the Patran database.

When the PATRAN - Advanced preference is set in the CSR program, the first time PATRAN starts, a new database is opened and the NASTRAN model is read in. A client data packet is written into the PATRAN database that contains the name of the NASTRAN file and the date and time the file was read into the PATRAN database.

The user now has two alternatives when the PATRAN session ends. The user can do a hard exit out of PATRAN or the user can just click on the CSR Stage 2 window and return to that program, leaving PATRAN running in the background.
When the user selects another instance where PATRAN is required, the CSR Stage 2 batch file looks to see if PATRAN is running. If it is not, things run as normal and a new PATRAN process is started. Only this time, PATRAN now looks for an existing database. If found it is opened and there is no need to read in the NASTRAN model again. If PATRAN is already running, then the batch file prepares the PATRAN input data but does not attempt to start another PATRAN process. The user must know to bring the PATRAN window to the front.

Each time other instances of PATRAN occur, PATRAN checks the client data packet for the appropriate data block to see if data already exists in the database. This prevents PATRAN from having to repeatedly read in the same data over and over every time a new PATRAN process is started.

A PATRAN pull down menu CSR Stage 2-AUI is available which displays all the various CSR Stage 2 entry points to PATRAN. All data blocks and the timestamps of when they were last read in are viewable using the CSR Stage 2 Datalink menu option. If the user has executed something in CSR Stage 2 that would change the data, the user can request to update all data, whereby PATRAN deletes each data block and reads in the data again.

There are several kinds of CSR Stage 2 data that could lead to a collision of data in the PATRAN database such as node IDs, element IDs, or loads that already exist in the main PATRAN database. To remedy this, when certain data need to be read in for a particular display that has nothing to do with the main model (such as panel buckling evaluation), the main PATRAN database is closed and a new database is opened to read in just the panel buckling information. When the user is finished with that PATRAN option, the buckling database is closed and the main database is reopened. And this all happens through PCL and the PATRAN process is never stopped.
Using the client data timestamps for data blocks and switching the main database with several ancillary databases gave the users the tool they needed to start PATRAN once and move freely between PATRAN and CSR Stage 2 without ever having to stop and start PATRAN.

**Conclusion**
This paper has illustrated many powerful PATRAN PCL programming innovations that were implemented to provide ship analysts worldwide with the graphics tools they need to do their job. As the analysis and evaluation requirements for safe ship design become more sophisticated, the model and results data become all the more voluminous. The Harmonised Common Structural Rules for Bulk Carriers and Tankers on the drafting board for 2012 is looking at model sizes of 500,000 nodes and 120 load cases.

A multi-national team of programmers and engineers successfully designed, developed, tested and implemented over 25,000 lines of PCL code to provide a powerful graphics interface between Common Structural Rules Software and PATRAN. The source code was maintained in an online Programming Version Control System (PVCS). When new code was ready for testing, a fresh mod of the PCL interface was compiled into a PCL library (.plb) and simply emailed as an attachment to anywhere in the world, where it was then used to replace the existing library and testing could begin.

PATRAN Programming Language (PCL) has shown over and over to be the preferred graphics programming framework for engineers throughout the world. Using the PCL functions, powerful data drilling and graphics display tools customized for ship analysis and assessment were developed. It is expected that further enhancements will be made to the PATRAN interface as the number of ship analysts using the new CSR Software continues to grow.

Although PATRAN is a powerful system when it comes in the box, the real fun, excitement, and fulfillment comes from thinking “outside of the box” … to imagine a new tool that would speed up an engineering process or create a new, useful display, and then write the PCL code to create that tool. With PCL programming you’re never limited by what PATRAN can or cannot do ... you’re only limited by your own ability to innovate and the drive to create something new.

----------

**About the Author**
Daron H. Libby graduated from the University of Maine with a B.S. in Civil Engineering. He then studied at Cornell University where he received his Masters of Science in Structural Engineering in 1982. Mr. Libby spent two years on the Space Shuttle Program at Rockwell International in Downey, California. He was the 13th employee in the Software Division at PDA Engineering in 1984, where he spent nearly ten years in PATRAN Interface Development, PATRAN Education, and Technical Support. In 1991, he founded DHL Consulting where he has provided expertise in finite element modeling applications and customized user-interface development for Fortune 500 companies throughout the world.

During the 1990s, Mr. Libby headed up a Bridge Team for the American Bureau of Shipping where he traveled with a group of American engineers to the largest shipyards in the world including Japan, South Korea, China, Taiwan, Poland, Italy, and Singapore.
PROFESSIONAL DIRECTORY

DHL Consulting
Daron H. Libby
Londonderry, New Hampshire
dhlc@comcast.net • (603) 425-2292

Michael Schmidt
Coppell, Texas
mschmidt2007@gmail.com • (469) 855-7787

CSR Software, LLC
Lloyd’s Register EMEA, 71 Fenchurch Street, London EC3M 4BS, United Kingdom
www.csrules.com • +44(0)20 7423 2211

Managing Director: Aidan O'Donnell C.Eng., MRINA
Ship Surveyor/CSR Domain Expert: Akizumi Fukushima (based in Houston)
Programme Manager: Jasbir Dosanjh