DBUG DataCAD Boston User Group
c/o Rick Gleason
114 Commonwealth Avenue
Boston, Massachusetts 02116

Mr. Livingston Davies and Mr. Peter Smith
CADKEY, INC.
440 Oakland Street
Manchester, CT 06040-2100

November 21, 1990

Dear Mr. Davies and Mr. Smith,

Enclosed please find the following:
1. Summary Spreadsheet of the DBUG Wishlists (with comments). 7 Pages.
2. Forty-nine completed DBUG WISHLISTS, August 8, 1990. (to Peter Smith)

We hope these responses will provide you with useful feedback on the program as it currently exists. I enclose these on behalf of DBUG, and Evan Shu and the others who have taken their time to complete the forms and send them back. I hope you will carefully review these items and consider them in your next release of the software.

The visit and discussion with Lou Bodnar, Clay Rogers and Mark White at the DataCAD User Group at Build Boston was certainly welcome. Unfortunately that meeting was not well attended (not unusual for DBUG Build Boston Meetings for some reason) but I'm sure whatever information gleaned will be adequately disseminated to the others.

This is the right time to discuss DataCAD's position in the market with respect to market share and technology. We have shown substantial commitment to the program over the long haul, and judging from the programs which are now available, DataCAD is long overdue for an overhaul and "great leap forward" (hopefully a real one). DataCAD has had a very strong position in Boston and a strong User Group, however **after seeing some of the programs available, I and others are becoming legitimately concerned about the competition.**

Ease of Use, a longtime DataCAD forte, is now being seriously challenged by a number of programs which we have reviewed. Several members of DBUG have seen Vellum on the Mac and find its "Intelligent Assistant" effective. I have had a chance to learn about ArchiCAD (also on the Mac) and have truly experienced "Interface Envy"! GOOEY WHIMPS and all. **Someone at CadKey should be studying carefully the way programs on the market work and what they do. This person should be a resource in the development cycle.**

The features of these and several other programs are worth looking at carefully for

ideas and techniques to improve the next DataCAD which I hope will be reincarnated in a new form which solves the following basics:

1. Platform and Operating System Independence.
2. More efficient use of memory on 80386 Intel based machines .
3. Use of the full 32 bit capacity of many machines.
4. New programming speed and flexibility to enable faster program development.
5. A parent database which supports all applications.
6. Application specific interfaces which use the database appropriately.
7. An integrated modular approach which allows CADKEY/DATACAD modules to be purchased with "mix and match" for various necessary applications.
8. Dynamic Linking to other programs for database/spreadsheet and word processing purposes.

We know that DataCAD 4.0 was a stopgap measure, evidenced by the major change being display list, additional drivers, features migrating into 3D, and a new manual. We have also heard rumors about "Killer CAD" and DataCAD 5.0 and 6.0 and more recently "Core Technology". We know Eric and others have been working for a number of years (admittedly interrupted) on a special advanced version. Many of us are hoping for a new "Glastnost" at CADKEY and some evidence of real progress because **programs that are now on the market are quite sophisticated.**

We also know that the initial Versions of OS/2 were not mature environments, and that the new version will be much better. We have read that Microsoft is now concentrating on Windows, with OS/2 and Presentation Manager being IBM's project. We understand Microsoft will be developing OS/3. Unix and X-Windows is still a real possibility, especially with graphics workstations prices, but who needs that big an operating environment?

I'd like to give you some idea of what **ArchiCAD** is like, because it certainly has some features worth emulating. (The Mac suffers from not having function keys, requiring separate mouse picks or alpha keys).

Library: Each library item has three components:

1. 2D/Ortho Plan Symbol Graphic.
2. 3D Symbol Graphic.
3. Database.

Symbol graphics are stored in GDL (Graphic Description Language) Files which are ASCII.

Parametric: The parametric part of the program allows selection of a library item, selection of the element of the library primitive entity to edit parametrically, and then a window shows the primitive (width, depth, height, radius, dimensions, etc) and

allows the user to edit any of the dimensions. In libraries where there are multiples of an entity, each entity has to be edited unless it is a sub-library.
Extensive Parametric Library of doors, windows, furniture, etc.

2D/3D: Plan representation and 3D views use the appropriate 2d or 3d symbol.

Doors and Windows: Doors and Windows must be a special case of Library because they have a very large set of parametric doors and windows with mullions, panels, different styles etc.
In plan, the window style is selected, parametrically altered for size, height, sill, etc. and effortlessly placed into the wall. The program seems to sense proper alignment with the wall plan and all linetypes (they call them pens) are automatically cleaned.
In 3D, the 3D window symbol is used, and when Shading is selected, the symbol for the window seems to automatically provide the necessary Void in the wall plane. There seems to be some intelligence between doors/windows and walls, so that defining the void is unnecessary. I did not get into wall thickness at all, or drawing a wall in elevation mode.

DataBase: Simple and easy spreadsheet type entry. Quantity takeoffs and reports of wall, floor, roof areas, materials, doors, windows, etc.

Lines: In plan, to each wall, you can attach line (pen) characteristics which are representative of building materials such as a typical wall (ie: Brick, cavity, gypsum, insulation, vapor barrier, gypsum with offsets). Each pen defines an entire wall section. By hitting a cleanup key all corners are mitred and doors and windows acknowledged. The user can define unlimited pens. In 3D the pen attribute is not shown. This feature is far better, and integrated than my SET-LINE macro for DataCAD.

Fill: A natural outgrowth of the above feature would be the ability to attach to various entities, fill and pattern attributes which would show in either plan or 3d (as determined by the user).
The fill patterns would recognize doors, windows, toilets,

and cabinetry and not draw the pattern behind or below. Perhaps the user would have to define which items to recognize in the initial version.

This feature would be great for creating a 3D shingle pattern on a roof or wall, or a tile pattern on bath elevations or floor.

Since it is a fill attribute it would not create discrete entities similar to hatch, thus saving drawing space.

This feature would almost be like texture mapping when rendering, only you could control which side of an entity received the fill attribute. Also you could control whether it would show in ortho/plan or 3d.

AutoPlot: This is the interactive solution. No drawing is required to be current. Takes a section of any drawing and allows plot at any scale, similar to an external version of 3D Views with *Lyrset and Clipcube.

Title Blocks are a part of this solution.

Ability to change lines, colors, settings, etc. - easier than editsets. Ability to change text sizes, fonts etc. Ability to add text notations.

Setup of an autoplot is saved.

Could also become a file and external layer manager perhaps. You need to see how this works! Also AutoCAD has "White Paper" which is similar, but I haven't seen it.

Shading: The program has shading features which is very useful for 3d visualization. On the old mac being used this seemed somewhat slow.

Cursor: Smart cursor which finds snapping points automatically and uses very interesting orthagonal and angle techniques. Similar to Vellum's Intelligent Assistant I think. There are many programs which have this kind of drawing assistance built in, such as IBM's cursor which dynamically shows key points and entities as the cursor is moved. More study on this would yield significant improvements in use, particularly number of command picks required, ease of drawing shapes at odd angles, ease of snapping to points, and if carried properly into 3d, ease of bringing a plane into the plane of the screen.

Windows: Popup windows seem to be the new interface technique. Inevitably DataCAD will be using them. Key elements are that the windows should be predictable regarding location, should also use function keys if possible, should permit user location if possible, and finally should not be constructed or sequenced so that the user is required to pick entities or locations in the main drawing which may reside behind the window! The windows should not rely solely on selection by a mouse.

Icons: I have not formulated a strong opinion on this yet. My ambivalence about icons versus text is probably due to my inexperience with Windows or Mac. However icons seem to relegate the function keys to oblivion, something I'm not sure I would be willing to give up, simply because a mouse pick often takes longer than hitting a function key.

So much for this brief and inadequate review of the program. You should see it and use it in the flesh. By the way, the fellow demonstrating ArchiCAD was an ex-DataCAD user from McGowen, Brooks and Reno in Maine! Enclosed are ArchiCad News and Features.

If you will bear with me, I intend to describe some improvements to the current version of DataCAD. These ideas are hopefully myopic, relative to the next release, but they should provide some kind of guidance regarding the kinds of tools we need. **Indeed, I hope you are programming even better solutions and that someone in the CADKEY/DataCAD organization is very familiar with the new programs and features in the industry!**

We need some smarter commands which bring a selected plane into the plane of the screen. We need verification that a certain plane is coplanar with the screen.

PlnrSnap: Currently use of clipcube and turning layers off is the best way to isolate and identify planes on polygons and slabs, etc. which we want to snap into the screen plane. We need a faster better easier way.

Snap to a point. The nearest plane of the entity found is identified in light grey. The following appear in the menu:
"Nextent" - Directs the program to search for the next nearest entity and change the snapping plane to the closest planar surface of that entity.
"NextPlne" - Directs the program to highlight the next planar surface of the current entity (whose planar surface is highlighted).
"*EntSnap" - This is a toggle which permits control of the

cursor snapping in the drawing window to be for the currently selected highlighted entity or for other entities as well. When the toggle is on, snapping is only to the selected entity. When the toggle is off, snapping is possible to the vertices of other entities, thus permitting the user to define a new plane between the snap points of several existing entities.

One of the problems of planar snap is that the user is never sure what entity is being snapped to, particularly when the vertices are close together. I find I am always zooming down to 12" scale, particularly when I am trying to enter VOIDS into an existing polygon or slab, or trying to add a vertice. The above features help solve this problem. I am sure that the command could be made to be smarter and more intuitive with more work.

It would also be great if once in "plnrsnap" you could move the cursor into the proximity of the plane you wish to snap into the plane of the screen, and the closest plane would automatically identify itself with grey blinking.
If that is the correct plane you then hit "enter", and the plane is automatically brought into the screen plane with the bottom of the drawing down, facing in the same direction as the previous iso view.

Identify: This is an alternative to ease snapping a given plane into the screen. Select 3D identify. Identify an entity. 3D entity is highlighted. Pick "plane" from the bottom of the menu and the closest plane of that entity is highlighted. If this is not the correct plane, pick "plane" again and the program determines the next plane on the entity. When the user has found the correct plane, pick "ScrnPln" to bring the highlighted plane into the plane of the screen.

Also when we use identify on an entity, it would be best if we were informed somehow that a plane of the entity is currently coplaner with the screen plane and also how far away. We need this for verification when entering new voids and vertices.

PlnrProj: We also need a 3D projection feature which will take a warped (or unwarped) entity and project it into the plane of the screen to reestablish a true plane.

Perhaps the feature would then confirm that one of the entities planes is in the plane of the screen. It would also be good to know somehow which surface is in the plane of the screen, or in the case of a slab how far in or out of the screen the other coplanar surface is.

Perhaps a macro that does all this would be a good start. You should be able to identify a plane on an entity, bring the plane into the screen, rotate the entire drawing to get the proper view (World), verify that the plane is in the screen, possibly project the entity into the screen plane (thereby translating the entity in world space). The macro would leave us at the 3d Menu.

Templates: It would be useful if the template windows could be brought up on a different monitor, and if multiple templates could be brought on screen at the same time. Details about how to drive two monitors and jump the mouse between them would have to be worked out, but this could tie into the need for reporting across multiple templates and multiple drawings somehow.

Text: The text editor seems to very high on the wishlist. One wonders about the difficulty of programming an advanced text editor with C++. Also with dynamic links, perhaps a full blown WP like Microsoft Word or WordPerfect is the way to go. The only problem with that is the implied associatity connection between existing text and certain graphic points via arrows or pointers. When the existing text is edited outside the program in a WP, it may loose its implied associativity with the graphic arrow.

Perhaps there is some way of making a "dynamic arrow" that is similar to a "new paragraph" key. In other words the arrows are dynamic and will point to the beginning of each new paragraph. Words wrap and text flows between lines within a paragraph. The paragraph is grouped together as a linked group of entities with a Text Paragraph attribute (ie Para #14). When a paragraph is moved the dynamic arrow moves with it.

One other feature is necessary for schedule generation and editing. Dynamic linking of schedules to the WP with easy editing in the word processor.

This might reduce programming time and redundancy, although with C+ + it might be quite easy to do within the program.

Database: There are endless discussions going on about whether the database should be separate from the drawing. I do not know or understand enough to be the judge of this. However there are very powerful database programs which should be dynamically linked to the program. Where the data resides is up to you guys, but we really need better spreadsheet type entry forms (flexible) and reports (flexible) which allow us to develop cost estimates, facilities management, quantity estimates, etc. The linking process should be intuitive and easy to accomplish so your users can feel confident about doing it. (Same goes for the WP).

Network: This will become more important, and the improvements we've listed are just a few of the basics. These can certainly be accomplished easily.

Linetypes: Review the wishlist suggestions. Linetypes should be developed similar to the the symbol system. When a user-linetype is selected or used the code is loaded into the drawing file similar to a symbol. In this manner a drawing can have unlimited linetype definitions and the user can store unlimited linetype definitions on the HD. Linetypes should be fixed so that they end properly and miter well. See the above description about ArchiCAD "Lines" or pens attached to walls which show in Ortho only.

Fill: This should be similar to hatch, but does not create separate entities. It should also load the fill pattern code into the drawing similar to symbols. It should be intelligent and know when it is hidden by other objects. You should be able to attach a fill pattern attribute to any face of an entity (ie Exterior Wall, Interior Wall, Roof, Ceiling, etc). You will have to be clever to make this easy for the user, so that laborious redefinition of edges or vertices is not necessary. The fill should recognize windows, doors, and voids properly. This is the precursor to shading.

Shading: A shaded view should be available in one window, for ease in understanding the wireframe.

Rendering: We also want a renderman interface. Velocity is good as far as it goes, but it doesn't support the shadows, lighting, texture mapping and other features that renderman does. **If you want DataCAD to really make use of it's 3D capabilities and remain a viable choice, we will need something with the abilities of a Renderman Engine.** After developing a "2d Rendering and 3d Animation" Seminar for Build Boston 1990 with David Pendery and Bill Riseman, I am convinced of this need. -DataCAD's strength has always been 2D/3D ease, now that rendering capabilities are clearly in the market, DataCAD must be competitive and **well connected in that area or its advantages as a 3D Tool will be lost.** Renderman Interface:

1. Viewmaster is a great program, it could be expanded to become the Animator for DataCAD.
2. A new DCAL macro that sets lights and assigns attributes to entities for Renderman textures and other routines. And then sends the translated renderman file out.

Perhaps you have some better solution.

Viewports: The three dimensional interface will be improved by multiple viewports or optional tile-type views which can be poped up to get oriented. Also I believe that a single shaded model viewport would enable better visualization.

ViewPlot: This part of the program should be similar to AutoCad's "white paper" and ArchiCAD's "AutoPlot". You should see how these work. Others have called ArchiCAD's program the best that they have seen. This is a part of the solution to "interactive drawings" and can also become a File and Project Manager" of sorts. See ArchiCAD "AutoPlot" above.

File Xfer: One final plea for a better file transfer mechanism such as IGES, with continued support of DXF. **DXF does not support voids or multi-vertice polygons. Which causes problems when trying to transfer our 3d models out to rendering and animation packages such as Crystal 3d and 3d Studio.** (the Renderman Engine with a DataCAD interface?)

There are many other things to review, the DBUG Wishlist is a good start.

There's a lot of work to do to stay competitive, and no time to hesitate, -make your best choice on direction and go for it. Do not get embroiled in useless and wasted political battles. The best programs should survive, **because a reasonably large share of the market will opt for quality programs.**

We would appreciate knowing what you are doing to keep DataCAD competitive.

Sincerely,

Rick Gleason

Rick Gleason
Co-Founder DBUG - DataCAD Boston Users Group

cc: Co-Founder-Evan Shu, Eric Smith, Dave Giesselman, Ken Parrish, Tim Dunne, Clay Rogers -Enclosures