FORM ABSTRATION
William Bricken
April 1989

Email correspondence for various sources.

> Alas, your abstract of abstractions has confused me even further.

Well, yes, we've been thinking about these ideas for a long time, and it does take experience with the perspective to make sense of my short comments. The best way to cover the concepts is to come talk one-to-one. That is, email itself is linear and necessarily compresses visual concepts. However:

> HOW does the user draw the abstract shape? I recently asked a similar
> question: How is cyberspace possibly useful for drawing, for example, lines
> and arcs?

FORM ABSTRACTION

the ability to refer to spatial and relational concepts without instances. Eg: the bay is outside my window, and it is full of boats. I can think of the bay and boats as an instance, or I can decouple the position of the boats from the space of the bay:

CurrentWorld = BaySpace * BoatLocation * BoatInstances

With the decoupled model, I can *multiply* to get the current reality, and I can change the components to get different versions of the current reality. By topologically redefining the BaySpace, I can have the same configuration of boats placed in a context of (for eg) currents of water (rather than volumes of water). I can change the instances of boats to instances of people-on-boats to get a world composed people at (boat) Locations in BaySpace. (Note that the map between boats and people-on-boats is not one-to-one.) Or I can work with Locations abstractly, by omitting BoatInstances from the equation.

To draw an abstract shape, it suffices to identify Points in a Space. Instead of boats on the bay, form abstraction simply allows Locations on the bay, without reference to boats, or to people-on-boats, etc.

Cyberspace means *direct interaction* with forms. If we are in 2D, we directly manipulate lines and planar figures. Cyberspace doesn't make *everything* easier, it makes it easier to directly interact with the forms native to a given abstract space. 3D (real) reality doesn't make points or lines easy either, that's why we use a projection of reality called paper.

Seems like the important question is: What structures and tools are most

natural for 3D space? Points and Lines aren't natural at all. We require CAD users to compose Lines into figures for the same reason we require programmers to compose words into figures: we just haven't reached the point of sophistication yet where we can use a more natural, higher-level language (a language of forms).

> How are CAD 3D objects and 3D object semantics differentiated?

With the current state-of-the-art, the *user* puts almost all semantics onto CAD constructions. We provide full freedom to draw in exchange for externalized meaning. Next, the CAD system itself will enforce meaning by *restricting freedom to draw*. Solid modelers don't have line primitives, cause lines aren't solids.

Semantics are the link between what you draw and what you mean by the drawing. I can draw a vertical line and call it a tree. Right now, nothing in the drawing knows that the line is a tree. Alternatively, I can enter a tree drawing regime, and draw the line. Now the automated regime knows that line is a tree. Since the tree regime knows about trees, I can merely select GROW, and the line will mature with branches and leaves.

Yes, someone has to do the knowledge engineering and programming to define the tree regime. That's the job of applications developers, to make tree models that look good, grow as we would expect, and drop leaves in the winter.

The Cyberspace project is building the tools that permit the applications developer to make tree models. These same tools permit the applications developer to make *programming models*, structures that act like programs. The line in this context might mean GOTO instead of SimpleTree.

The hard part of making abstract tools is to assure that they are *composable*. Not only must the line have a consistent meaning within a regime, combinations of lines must also have an intuitive meaning. Assuring meaning, composability, and consistency is the art of modeling. With Cyberspace, we are building a general modeling capability. You furnish the knowledge, we furnish to tools to make that knowledge manifest.

The first application of Cyberspace is with architectural drawings. You provide the meaning of building, we provide the tools to treat the representation of a building as if it were a real building. Obviously we can't be completely convincing, but one of the discoveries of cyberspace research is that human minds require little in the way of representation clues to believe the as-if-real hypothesis. Another result is that consistency and detail are necessary to maintain as-if-real.

The second application of Cyberspace comes quickly. Since the building model

is parameterized, it can be abstracted and reconstructed in a different world of instances.

You want to walk on water?

(substitute BaySpace for FloorSpace in Building)

You want to float around?

(substitute Boat for Person in BuildingSpace)

Want to fly?

(connect 6DegreeOfFreedomFlyingControls to Person in Building)

Want a city?

(multiply CityBlockForm by Building)

We are building Substitute, Connect, and Multiply, and the definition substrate to easily specify what Bay and Floor and Building and Boat mean.

Comment (designers)

The potential ambiguity of the 3D world is addressed by careful context design, in which multiple orientation and depth cues are included. How would 2D representations be useful in Cyberspace? You can't get inside them...

When moving among 3D objects, reaching out to turn one of them around with your hand makes more sense than any indirect method. Todd was NOT in Cyberspace, he was manipulating an object represented in two dimensions with a device intended for 3D interaction.

Reply (William)

Cyberspace is not solely a 3D idea. You can get inside 2D shapes, if you have a 2D mind. We talk about 3D most because it's the frontier, and our bodies are there. But 4D is around a trivial corner. We can change *scale* as well as x,y,z. Animation offers another dimension, we can play with time too in Cyberspace.

I expect we'll be in 5D before the end of the year. With many more to come in the next decade.

An Initial Reaction (management)

What's everyone talking about? Form abstraction? Spatial and relational concepts without instances? Multiplying decoupled models to get realities? 3D being more ambiguous than 2D? Identifying Points in a Space? Forgive me, but isn't this all just Pure, Unadulterated, Bullshit?

One of the reasons for developing Cyberspace in a CAD context is to make things easier - we live in a 3D world, therefore it should be intuitively obvious that working *directly* in a 3D design space should be easier and more understandable than poking at wireframe models on a 2D screen. One of the big reasons that 3D CAD systems aren't as popular as 2D systems is that it is HARD to work on 3D models when you're peering at them (generally non-dynamically) on a 2D screen. The reason that Silicon Graphics has been successful is that they make this easier with fast hardware. I'm not sure how "multiplying and substituting forms" makes things simpler.

I'm also not at all convinced that it's easier to manipulate objects in Cyberspace by mapping my hand into it with the DataGlove. My hand, being connected to my arm which is in turn connected to the rest of my body, isn't capable of moving in all six degrees of freedom. Why limit yourself to interactions that can only be performed with human appendages; I don't have that restriction when I work with CAD systems today. But that's not important; there are many ways of specifying all six degrees of freedom in a 3D space, I (and tens of thousands of other people) do it every day using a 3D CAD system.

Put it this way: building objects in Cyberspace can certainly be no WORSE than building objects with any of today's 3D CAD systems - all of the same tools can be made available if that's the only way people can understand how to use it (another reason why 2D CAD systems are still more popular than 3D). But it has the potential to vastly IMPROVE the interaction by allowing the user to directly interact with the model and space.

Look, all it is right now is a viewing pipeline, like every 3D CAD system in the world has, coupled with fast hardware and 6 D.O.F. input devices. That's all we've got, and that's all anybody in the world has at this point. How to apply this in a way that both improves the way we do design (and many other things) and is understandable by a large enough body of human beings to make it popular hasn't even been approached yet. But it's almost certain that Cyberspace is a better way of doing things if for no other reason than the fact that people like *interactive* environments better than batch environments. It is for this reason alone that I believe this is an inevitable future.

Get Comfortable with Abstraction (William)

Gee, you really should get comfortable with the abstract tools and the architecture of Cyberspace.

I fully agree with you that *to the user* Cyberspace makes things easier. And that is a sufficient reason for its value.

But what we are talking about in email is the architectural and design details. You are totally wrong that

> all it is right now is a viewing pipeline, like every 3D CAD system in
> the world has, coupled with fast hardware and 6 D.O.F. input devices.

That is what the hardware is. The software is much deeper, and includes abstraction hierarchies that permit articulation of bodies, multi-body interaction, and physical modeling. In general we are building modeling tools that are relatively independent of the hardware substrate. Multiplication and substitution of forms are the mathematical techniques that permit Cyberspace to be easily configured.

If form abstraction and decoupling are bullshit, then so are matrices and vectors. Both matrix algebra and form abstraction are mathematical techniques for manipulation of geometric structures. What would motivate you to call our formal architecture bullshit? How would you design a Cyberspace Construction Kit that permits arbitrary models? How would you, for example, introduce gravity into CAD drawings?

Non-numerical multiply (william)

> It's a very small point of terminology, but every time you say "multiply",
> I go through the mental hiccup of saying to myself "no, he really means
> 'compose' or maybe 'dot'". The asterisk symbol doesn't seem to throw me,
> but the word "multiply" is inextricably tied to arithmetic in my mental
> space....

Yes, you are totally correct that the * operator on a drawing is function composition. But just like the concept "multiply" is used broadly (ie is overloaded) for matrix operations, I (in my personal mental vocabulary) overload the concept to mean "compose". It's a conscious decision. I'm trying to elicit the image of a "field" (a closed system under two operations named + and *). In the theory of matrices, we could interpret a vector as an operator, and "dot" as function composition, also. It's a free choice to think of the {x,y,z} structure (a point) as an object being multiplied, or as a function being composed. What I wish to avoid is the image that + and * (whatever the name) are not related. They form (via distribution) the exact same organization that plus and times (of integers) form.

The more interesting idea (to me at least) is that a turtle specification,

"ftftftf" for example, is both a function that composes with other turtle specs, and a data structure that can be multiplied.

And as a bit of context, I can show you an isomorphic definition of integers that uses function composition (ie is implemented via substitute) to achieve "multiplication". That is to say, an integer (5 for example) can also be *read* as an operator as well as a data structure.

So it comes down to this: calling * either "multiply" or "compose" gives instructions to the reader to interpret objects as inert or active (respectively). Mathematically, the two are the same.

Re: Abstractions... (management)

All of the "abstraction hierarchies" you mention for articulation of bodies, and multi-body interaction are available RIGHT NOW in CAD, COMPLETELY independent of hardware platform. I use all of them for making animations, including simulating gravity and non-linear interpolated deformations. All of this is done with existing tools and a few hundred lines of code.

I never suggested that the "formal architecture" of Cyberspace is bullshit. But little of it exists yet, and you were speaking as though it did. We have bodies with inherited transforms. The bodies can be in motion, defined by a motion vector and function. That's what we have. If we can do more with the current implementation (like multiplying to get the current reality), I haven't seen it demonstrated nor heard about it in any of the progress meetings. We've barely started on the hard problems of picking, fast rejection and collision. I'm not knocking our progress, just trying to make sure we know what we have and aren't telling people we have more. If I'm wrong, show me.

I would also suggest that many of the things you talked about regarding form abstraction and substitution can be had TODAY. With the added benefit of having a tool to construct the models. I'm still not sure how one creates forms in Cyberspace. How would you create the boat hull on which to put your people with which you multiply the two?

Reply (William)

A couple of observations:

Designs and architectures can (and do) *exist*, without being implemented. The formal architecture exists in both the design notes and the code.

In addition to bodies in motion (with inherited transforms), we have bodies reacting to arbitrary configurations of forces, and bodies acting as

autonomous agents (with localized data, processes, and dispositions). We have control and database parallelism, type hierarchies, component hierarchies, control and coordination hierarchies, and event-driven responsiveness. We have designed abstraction barriers for multiple representations, abstractions for storage management (in parallel), abstractions for process management, and computational regimes. We have algebraic and logic deduction (over classes as well as instances). We have structural decomposition and arbitrary metric and non-metric spaces. This stuff forms a cohesive and extensible design (and is only partially implemented).

(This one is important) The issue has never been providing *new* capabilities that CAD can't do. Instead, the question in my mind is "How do we build an integrated, intuitive modeling system?" Attaching Cyberspace to a CAD system through interface programming is not the way to get real-time performance.

You are free TODAY to build Cyberspace models in CAD. The question is not construction (cause CAD does that), it's modeling. A model (this is a technical term) is a structure that permits operations over all possible instances of a class. I suspect our difficulty is one of terminology. We need a Construction Kit that lets me place a gravitation field in a space, so that the objects constructed within that space obey the law of gravity.

Reply to management (cyberspace programmer)

I agree with you that the interactive aspects of cyberspace promise to be a big step forward over batch approaches, but it's a lot more than that too. You asked what we're all talking about. Well, we're talking about what cyberspace CAN BE. It's true that what you've seen to this point is basically a viewing pipeline coupled to some fancy hardware, but you're only seeing some very early results. A viewing pipeline is certainly fundamental, but it never seemed to me (or, I daresay, to William) that we were endeavoring to create a viewing pipeline. Rather, I took the viewing pipeline as a given, something essential to be fitted into an architecture that would foster the evolution of cyberspace. The focus has always been on the architecture and, most importantly: the ultimate potential of cyberspace. If you think we're merely hacking out a viewing pipeline, then we really need to talk, because we aren't on the same wavelength.

I also agree with your observation that "how to apply [cyberspace] ... to make it popular hasn't even been approached yet." But then, I thought that was our job. Now, I must admit, I'm more than a little confused. I just can't fathom what sense it makes to assemble a group of passionate, creative people, ostensibly to invent a grand new way of doing things, and then to censor them as soon as they begin to put their best feet forward. I've seen a lot of real bullshit in my time, and I've also seen what I thought was "bullshit" turn golden. How can you expect to invent lucrative new products

when you jump to shut off the creative mill as soon as it starts?

I'll be honest with you: I don't always understand what William is talking about, but I don't jump to the conclusion that what he's saying is bullshit. Rather, I give him the benefit of the doubt and figure he's tuned into something I'm not. The important thing is that he's out there exploring, inventing, and doing his best to question and knock down the established order. That's what R&D is all about.

In my opinion, William is an eloquent spokesman for cyberspace. Whether I agree with him or not (and I don't always), I know that he's trying to conceive of something that has never existed on this planet, and I figure it's my job to help him do that. The least I can do is listen and encourage him to press on, to make the ideas clearer and clearer so that, finally, they can be realized. It's not just that I happen to like William. It's simply what colleagues do for each other.

Really, do you want just another interactive 3D system from the Cyberspace Project? If that's all it amounts to it's going to be a real tragedy, because we can make so much more of it. It's in no one's interest, least of all yours, to sell the project short. I know you're trying to be constructive, and are basically just asking us to be realistic. But please notice that we ARE being realistic. We aren't just waving our hands here. We all take this project very seriously and consider it a great privilege to be involved. e are driven by a vision of a new, more natural way of interacting with computers, and are seeking to implement the vision with a lot of attention to nuts and bolts issues. Remember that we set out to draw attention to cyberspace, to attract hardware manufacturers, and establish us as the leader in a new industry. We aren't trying to replace CAD systems. Rather, we're trying to complement CAD with other capabilities, and thereby to open new markets. And man, it's happening! The word's already out that this is the place to be, with regard to cyberspace, and it's just the beginning. I mean, c'mon, let's not snatch defeat from the jaws of victory.

So far you've only seen the viewing pipeline. Why don't we get together and talk about the architecture, and what we're trying to achieve? I think you'll find that it's all been laid out rather carefully, with a view toward capabilities that go way beyond today's desktop approaches.

Another technical opinion (CAD developer)

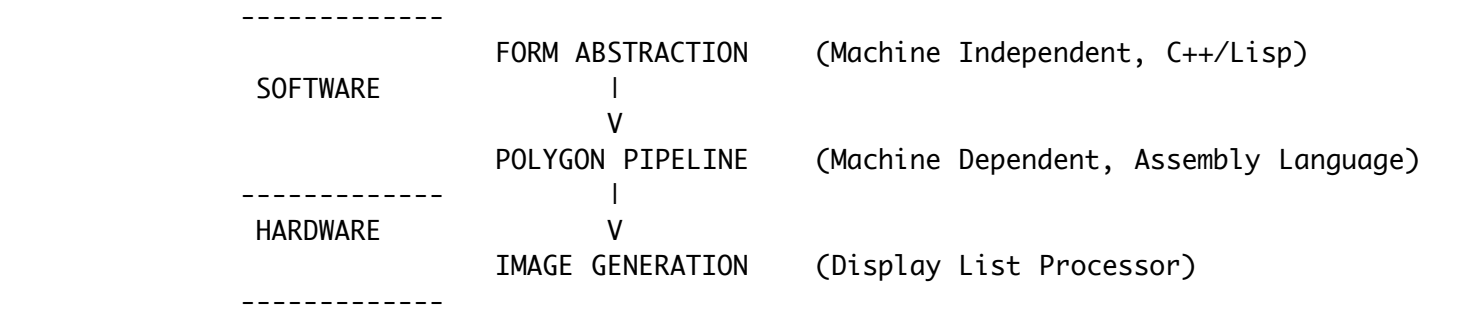
At one end of a possible cyberspace formalism is form abstraction. At the opposite end is the display. How is abstract geometry linked to image generation?

Probably the quickest way to convert from form to image is by making polygons. [Ray tracing is certainly out of the realm of real-time, even

using the AT&T Pixel Machine.]

We're talking Gouraud shaded polygons. Maybe even with smooth shading.

If forms must be reduced to polygons, the cyberspace formalism probably needs to include a "polygon pipeline." Thus:



This might seem trivial. But there are several questions that can arise from this simple model:

1. Is the polygonal representation going to be the ONLY real-time representation over the coming years?

2. Can the pipeline be generalized for other image representations?

3. Can form abstraction be compiled into a machine dependent set of data? Is this useful?

4. Are display list processors going to be competitive with MIMD architectures in the future? If not, can the pipeline approach include hooks for parallel processing?

5. Does there even NEED to be a pipeline? Why?

6. Does there even NEED to be form abstraction? Why?

I'm not suggesting that these questions need to be answered, or that this hierarchy is in any way valid. But it may be useful to start formalizing at this level, and then to build from it.

[Sidenote: Doesn't hardware become as important as software when dealing with the constraint of REAL-TIME?]

Management

My apologies to the group. My attempts to try and make a complex subject

simpler only served to piss off the people working on the project (the last thing I intended) and showed my lack of information on the design that we have.

My comments on our "only having a viewing pipeline" come only from what I see today and were apparently totally off base with respect to the design. I plead stupid and ignorant (and also plead that we discuss that design - the "basics" - in this group to help solve my problem!).

Please ignore further criticisms from me - constructive or otherwise - until I can understand this. Please also understand that the Cyberspace project is the most important project in this company to me (second only to getting out the next rev of our product so that we can all be paid to continue it...).

William

Your comments and perspective are valuable and valued. I'm particularly happy to see your reminders that CAD is central to our vision.

I fucked up not providing you with more design details. But designs are transients...

Sure makes me feel good that you value to project.