ELSEVIER

Computer-Aided Design 37 (2005) 461–470

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Ten challenges in computer-aided design

Les A. Piegl*

Department of Computer Science and Engineering, University of South Florida, 4202 East Fowler Avenue, ENG 118, Tampa, FL 33620, USA

Abstract

This short paper presents 10 challenging research areas in the general field of computer-aided design. The research problems come from the author's personal experience, and as such are highly subjective. All findings and opinions are those of the author and do not represent any of the institutions the author is affiliated with.

Keywords: Computer-aided design; Research challenges; State of CAD status assessment

1. Introduction

In 1905 Albert Einstein published three remarkable papers in *Annalen der Physik* [1]. The first one contains material on the photoelectric effect for which he received the Nobel Prize in 1921 (the relativity theory was still too much to digest both scientifically as well as politically). The second one *On the electrodynamics of moving bodies* forms the basis of what is known today the theory of relativity, and the third one, only three pages, is the first step toward the famous equation $E=mc^2$ published in 1907. Physicists regard 1905 the *miracle year*, however, Einstein would probably disagree. He arrived at his findings not by miracle but by hard work and stubbornness. Or as he put it [2]:

> "*I have little patience with scientists who take a board of wood, look for its thinnest part, and drill a great number of holes where drilling is easy*."

Throughout his life Einstein looked for the thickest part of any wood he could find and drilled his holes relentlessly. He dared to rebel against scientific as well as political opposition to pursue, what he believed was, the right science. He was not pressured by administration to publish; he worked on solving problems and not on polishing his resume.

Today's scientific landscape is quite different. Academics deliberately avoid the thicker part of the wood, and if they encounter challenging problems, they give up in a heart beat to move on to other problems that are more likely to produce publications. Unfortunately, this kind of science leaves numerous challenging problems behind, some of which cut into the very guts of commercial CAD systems.

In this paper I would like to invite the reader to visit some of the thickest part of the wood, representing research as well as implementation. The problems presented herein may require a decade or so (stubborn) research, i.e. large number of papers may not be on the horizon. However, if reasonable solutions are found, the impact of the contribution is well worth every bit of sweat.

2. Challenge no. 1: robustness

By far the most challenging issue in CAD is robustness. While one can cheat the eye in computer graphics and animation, the milling machine is not as forgiving. Some of the relevant problems to be dealt with are discussed in the subsections below.

---

* Tel.: +1 813 974 5234; fax: +1 813 974 5456.
E-mail address: lap@cadconferences.com.

0010-4485//$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2004.08.012

2.1. Proper metric

Anything in CAD is the matter of measurements. The questions that may need deeper investigations are:

- Would one universal metric fit all applications? The answer is probably no.
- If there is no universal metric, then how many systems do we need and what are they?
- How to change the metric from one application to the other without significant loss of accuracy, detail or meaning?
- How to mix different metrics, e.g. absolute length and relative brightness, within one context? For example, what does it mean: use seven feet distance after dark?

2.2. Proper tolerances

During CAD processes one uses a myriad of tolerances, many of which are not directly related to the actual manufacturing process. Some interesting questions here include:

- What are the most relevant machining tolerances?
- How to set the army of computational tolerances, e.g. those of systems of equations, to guarantee machining within the required accuracy?
- How tolerances in different spaces, e.g. in model space and in parameter space, are related. That is, how to set the parameter space tolerance in order to guarantee model space accuracy?
- How tolerances in different arrangements, e.g. parallel and perpendicular lines, are related? That is, should we set different tolerances for parallel and perpendicular lines, and if yes, then how to relate them to one another?
- Should there be one application tolerance, or should it be work space related?

2.3. Robust algorithms

Algorithms are the places in any CAD process where the inherently ‘dirty’ data, e.g. point cloud, gets received and converted into another set of data, e.g. an STL file, for manufacturing. The issue of data processing will be looked at later, however, what must be understood at the outset is that proper algorithm design begins with the selection of the right solution. The most relevant requirements for the right method are:

- it must be easily *understood*;
- the *implementation* must be reasonably straightforward;
- the results must be *accurate* and *correct*;
- the algorithm should lend itself to proper *testing*; and
- *maintenance* should not require a lot of time and effort.

If a method does not satisfy the above requirements, it is not suitable for CAD applications. Once the proper method is chosen, the issue of algorithm robustness comes into play:

- robust implementation: all correct input must produce correct and sensible output;
- data validation: data must be validated before entering the code and before leaving it;
- proper input/output handling; and
- proper data format conversion.

2.4. Numerical instabilities

Numerical instabilities account for the majority of computational errors in commercial CAD systems. The issues that may require some more investigation are:

- the selection of the right mathematical formula, e.g. power series versus spline formulation;
- the selection of the right numerical technique, e.g. elimination with or without (partial or full) pivoting;
- the handling of round-off error and the accumulation of error;
- the choice of the appropriate blend between degrees, smoothness, speed, etc.; and
- the handling of the noise inherent in the input and the intermediate data.

The problems related to robustness haunt every programmer who has ever worked on commercial systems. Fixing numerical bugs can be very frustrating, and often times results in patching up the code simply because no solution exists to remedy the problem. Current efforts in interval arithmetic and fuzzy logic may look nice, however, they may open up new problems in the process of solving some old ones. What would be a significant help is to know as much as possible about the entities to be computed on. For example, if we know how far we are from the root, i.e. where is the guess point in relation to the root, then the method can be adjusted to guarantee convergence. The term *knowledge-guided* computation is used to describe this kind of process. That is, every piece of data comes with a knowledge base to guide the computation, e.g. arrangements of lines come with a knowledge base that flags lines that are meant to be parallel, perpendicular, overlapping, etc. History has shown that blind computation can fail miserably leaving CAD systems vulnerable to break downs.

3. Challenge no. 2: geometric uncertainties

Although geometric uncertainties are related to robustness and tolerancing, there are a number of extra issues well worth deeper investigations.

3.1. Special cases

Geometric arrangements are full of special cases. The most notable ones are:

- cases of touch, overlapping, containment, etc.;
- cases of parallelism, perpendicularity, coincidence, etc.;
- cases of symmetrical data, data clustering, dense or sparse data, etc.;
- cases of degeneracy, discontinuity, inconsistencies, etc.; and
- problems with cracks, excess material, lack of detail, etc.

In just about any code that deals with geometry, the number of special cases is significantly larger than the general ones. Although writing special code for these cases enhances the robustness of the algorithm immensely, one must guarantee the robustness of the special code which, in turn, can make the algorithm much more difficult to maintain.

3.2. Data explosion

Data explosion is the result of careless selection of the methods, e.g. parameter space-based sampling, and improper implementation, e.g. recursive algorithms. Some of the relevant issues are:

- sampling: over sampling, sampling in incorrect places, etc.;
- procedural definitions, e.g. lofting a large set of curves may result in an explosion of control points;
- excess data on input may get magnified further to fill available memory;
- improper data structures, e.g. arrays of fixed size holding very little data; and
- non-compacted data bases used for further processing.

A real dilemma is what to do with the data on input. On the one hand, data reduction and cleaning seems like an absolute must in the actual CAD practice. On the other hand, proper data output should contain compacting and cleaning the result so that the receiving process may *assume* that the data is clean and is free from excess material. Until the CAD world is not willing to follow this common sense rule, clean and compact data cannot be assumed, I am afraid.

3.3. Data validation

Ideally all data that get passed around should be valid. Unfortunately, we do not live in an ideal world and data validation becomes a critical issue in all CAD processes. More precisely:

- data must be validated on input as invalid data not only generates nonsense output, it can, and often does, crash the system;
- data must be validated before computations begin to ensure that unpleasant things like zero division do not occur;
- the newly computed data must be validated before the output is generated; and
- all data must be validated before export or import operations.

3.4. Geometric beautification

Although CAD processes are supposed to produce valid and 'made to order' models, the reality is that most (if not all) models are rough and require post-processing, i.e. beautification. Some of the most frequently needed tasks are:

- removing unwanted edges, corners, cracks, etc.;
- removing bumps, oscillations, curvature extremes, etc.;
- healing incorrect models, e.g. removing holes in triangulations; and
- smoothing, fairing, re-shaping, etc.

The other relevant issue is to find the right mathematical formulation for aesthetics, e.g. what is the quantitative measure of flatness, curviness, smoothly changing, fairness, etc.

4. Challenge no. 3: error handling

It is commonly known that

> "*Bravery is not the absence of fear, but rather, how well one handles it.*"

Following this age proven principle one could safely say that

> "*Robustness is not the absence of errors, but rather, how well the system handles them.*"

There is no system without errors, in fact, at almost every single line in the code some error can creep in. Unfortunately, very little (if any) research is being carried out on how to handle errors in CAD systems, or at least very little is being reported. Finding the right solution is just as important as dealing with the error, in fact, because the solution is computer dependent, handling the possible errors becomes more challenging than handling the original problem.

4.1. Error prevention

The best way to deal with errors is to prevent them from happening. Of course, it is easier said then done, however, there are a few things that can be (and should be) done to minimize errors:

- data analysis and cleaning before computations begin;
- extracting and handling special cases to avoid numerical problems, e.g. a line-circle intersector can tackle the no

intersect and touch cases geometrically and uses algebraic methods only for the general case;
- choosing the right form of computation, i.e. some methods are faster but error prone, others are slower but less prone to errors;
- guarding critical steps, i.e. at each important step, e.g. division, in the code thorough error checking is performed before the computation is carried out; and
- data preparation for proper output, e.g. cleaning and compacting the output data.

4.2. Error detection

Once all measures are taken to prevent errors, the next step is to detect their possible occurrences. Some ideas that may be used are:

- developing an extensive error checking mechanism, e.g. special purpose code can be written that looks for errors;
- pre- and post-operations checking, e.g. checking the data before and after operations to look for inconsistencies;
- check or not to check, i.e. not everything can be checked and therefore the question is what part of the code can be left unchecked; and
- speed versus safety, i.e. error detection comes at the price of speed (and storage) so how much speed can be sacrificed without significant loss of performance?

4.3. Error correction

Finding errors may be difficult, however, correcting them is yet another challenge. The current practice entails the following:

- report the error and return control to a higher level code;
- correct the error (if possible) and continue the operation, assuming the input data has not been changed;
- ignore the error (not relevant or not serious) and continue; and
- document the error (in an error data base) and return, continue or halt.

4.4. Error documentation

Errors are like career criminals, i.e. if not dealt with properly, they keep coming back for more victims. Therefore errors must be documented properly for a long term error handling strategy. Some ideas that have been used are:

- the application of pre-defined error indicators;
- building an error tree according to the calling mechanism and recording the cause and the place of the error; and
- if space allows, a complete error history can be built with as much information as needed.

A very relevant, and often times annoying, issue is the type of error messages used by commercial systems. Messages like ‘An error has occurred! OK.’ is quite common. No, it is not OK, especially when the next line reads ‘Tell Microsoft about it.’ There must be a more intelligent way of dealing with errors, other than ‘telling Microsoft about it’ or shutting down the system.

5. Challenge no. 4: CAD engines

The time seems to be just about right to support CAD processes with special purpose hardware, i.e. casting CAD into silicon. This would have at least two major implications: (1) all processes could be accelerated, and (2) new solutions could be devised to solve problems difficult to handle with software only, e.g. surface–surface intersection.

5.1. Special purpose hardware

At the very minimum, special purpose hardware pieces could be built to serve particular sub-processes. A few areas worth mentioning are:

- support for special computations, e.g. support for numerical processes in hardware;
- support for CAD visualization, e.g. speeding up NURBS rendering;
- support for data input and output, e.g. haptic devices; and
- support for manufacturing, e.g. hardware for slicing STL models.

5.2. The CAD chip

The next step would be to cast major components into hardware. More precisely:

- casting libraries into hardware, e.g. geometry library;
- casting critical computations into hardware, e.g. surface fitting to point cloud;
- casting critical methods into hardware, e.g. subdivision; and
- special purpose chips, e.g. a chip to accelerate collaborative design.

5.3. The CAD work station

Once important chips have been built, the CAD work station can be designed. The work station could have components like:

- architecture for special tasks, e.g. ray casting for solid modeling;
- special purpose CAD units, e.g. the geometry chip;
- special purpose memory arrangements, e.g. memory for recursive subdivision;

- support for CAD data interfacing, e.g. IGES or STEP; and
- support for the major elements of the production process, e.g. NC machining.

5.4. Hardware bugs

A major trouble spot in CAD hardware is hardware bugs, i.e. how to debug them? CAD processes are full of error spots, so if they are cast into hardware, how to deal with them? That is a real challenge indeed and I can offer only a few vague ideas:

- proper error handling in software before casting into hardware;
- designing error correcting code, a real challenge indeed;
- developing several solutions to the same problem and casting them all, and if one fails, a switch to the next one is made; and
- hardware replacement, i.e. bugs must be fixed and the hardware is replaced.

6. Challenge no. 5: 3D

It may come as a surprise to the reader that studies done by the author and his team indicate that the majority of the population is 3D blind! That is, the majority of the subjects have been found to have difficulty in seeing and visualizing in 3D. The reason why the majority of design tasks are still done by paper and pencil is not necessarily historical but has a serious biological component to it: people are much more comfortable with 2D than with 3D.

6.1. Direct input in 3D

Traditional CAD design involves the input of 2D entities, e.g. interpolation points, a shape design in 2D, e.g. a profile curve, and a 2D to 3D design tool such as an extrusion or a revolution. Several questions emerge:

- is direct 3D input really wanted by the designers, and if yes, can it be made natural to human working styles?
- can 3D input devices be made easy to use without wearing 5 pounds of equipment or holding a light pen in mid-air for several hours?
- if we are so much attached to 2D, can 2D devices be used for 3D direct input? and
- what would be the major ingredients of an interface that supports direct 3D input either with 2D or with 3D devices?

6.2. Direct design in 3D

If we figure out the technology for natural direct 3D input, then the next step is to design methods for direct design in 3D, without going back-and-forth between 2D and 3D. A few interesting research problems are as follows:

- would designer really want to design directly in 3D, or would they still prefer to be attached to the paper sketches?
- what are the devices most suited for direct design?
- what are the software tools most suited for direct design (and styling)? and
- how does one go about developing an interface for direct 3D design?

6.3. Engineering drawings

Engineering design requires sketches as well as drawings, commonly referred to as engineering drawings, taught in most engineering colleges. The few challenges here include:

- standards for 3D drawings;
- standards and methods for projections;
- methods of 3D reconstruction from engineering drawings; and
- standards for 3D labeling.

6.4. 3D Displays

One of the many things that makes 3D design difficult is the fact that current displays are all 2D. That is, all images are projections onto a flat panel. Interesting projects in this area may include:

- creating the illusion of 3D via 2D devices, e.g. stereo projection;
- applying 3D to 3D projection to enhance the ability to visualize complex scenes, e.g. theatre perspective (also known as Burmeister projection);
- developing tools for 3D hardcopy, e.g. the 3D plotter or printer; and
- developing new display technologies, e.g. vapor based displays, for 3D display.

7. Challenge no. 6: from idea to design

In the 1960's, during the development of SKETCHPAD at MIT, S. A. Coons gave an interview to a science reporter. In it he introduced the computer as a human assistant and said that the team was working on tools that are 'natural to the computer and natural to men.' Looking at CAD systems, developed in the past 40 years, it appears that the 'natural to men' part has been swept under the rug. It takes literally years to master a CAD/CAM system, and once a designer becomes comfortable with it, it is unlikely that he will want to change to another one, even if this one is more powerful and has lots of extra capabilities. Any half way competent designer can transfer his ideas onto paper within minutes, however, conveying the same idea to another designer via a CAD system may take hours, days or even weeks. The core of the problem is that developers first invented the mathematics of CAD, and then wrapped some code around it to create the design tools. Naturally, the opposite would have been desirable, however, many commonly used traditional design tools may not admit easy mapping to mathematics or computer tools. Although the process from idea to the actual design, using a CAD system, remains clumsy, a few things deserve particular attention.

7.1. Human cognition

During the development of a CAD system the human should not be left out of the equation. Several issues could be looked at:

- 3D blindness is a major problem and tools must address this issue;
- computer tools should consider human cognition, in addition to what is feasible mathematically;
- people learn much faster by seeing examples, that is, CAD design tools can be enhanced by design-by-example methods;
- design is a learning process, i.e. one learns about a particular design as he goes through the various stages, which would strongly suggest tools based on learning.

7.2. The CAD psychology

CAD systems are used by humans not by robots. That is, individual differences must be considered. More precisely:

- there are differences in visual perception, i.e. we all see different colors, shapes, depths, etc.;
- there are differences in 'inner' CAD tools, i.e. tools must consider the psychological differences among designers;
- design (art) is a form of self-expression, and each CAD system should support that by allowing the user to customize the system; and
- psychological studies should always be done before system building begins to account for the human element.

7.3. Human factors

Since the advent of computers scientists discovered the human element in computing, which has become a major area of research in the past few decades. There seems to be no research in human factors in CAD. Some interesting issues that could deserve more attention are:

- individual preferences, e.g. some prefer designing through sketches other by examples;
- cultural differences, e.g. designers' working styles may be quite different in India, China or in the USA;
- gender differences, i.e. it may be surprising to realize that male and female designers do not think alike and may require different tools to accomplish the same task; and
- level or preparation, e.g. CAD tools must consider the level of education, experience and the type of experience of the user.

7.4. CAD system made easy

It is not uncommon to see employees being trained on a new CAD system for several weeks before they can take on any assignment. It may take an additional six months to a year to become comfortable and productive with the system. What is even worse is when the company changes from CAD system A to B, the training has to be repeated. This is like taking a new driving course to learn how to drive a Toyota after driving your Honda. One should be able to hop behind the wheels of a new CAD system, and after adjusting the seats and the mirrors, the designer should be able to take off. After driving around the country side for a day or two, your new CAD system (Toyota) should feel just as comfortable as the old one (Honda). A few ideas that may be worth pursuing are:

- developing CAD systems with a short learning curve, i.e. instead of focusing on the bells and whistles of the system, developers should focus on how easy it would be to master the use of the system;
- following the basic principles of Coons, i.e. find CAD solutions to problems that are natural to both the computer and the user;
- creating tools for the design practice and not for the mathematics involved, i.e. studying how designers work, and trying to map their working styles to mathematical and computational tools; and
- revisiting SKETCHPAD, i.e. it sometimes pays to retreat and go back to the drawing board to see where things got derailed.

8. Challenge no. 7: CAD for the disabled

How many times have you been told that if you want to see something better, you should close your eyes? How many times have you witnessed that people with disabilities had enormous artistic and creative abilities? Unfortunately, the world of CAD seems to belong to 'normal' people and folks with physical or psychological challenges do not have a chance to participate in the creative process. I, for one, would be very interested to know what exactly a blind person sees, if he could have the tools to express himself.

8.1. CAD for the blind

Probably the worst disability in CAD is blindness, given that most CAD systems require the user to see, i.e.

practically all feedback is visual. I personally believe that if blind people could use computer tools to show us what they see and how creative they are, the world would be amazed by their work. Some relevant issues that should be considered are:

- blind people have enhanced perceptual senses, i.e. just because they are blind, it does not mean they cannot see (sense the world around them);
- they have a sophisticated 'inner' CAD system, i.e. they can visualize what other people may not be able to;
- blind people tend to have a high level of creativity and artistic ability; and
- they have an enormous drive for self-expression.

The key principle here is what Albert Einstein so elegantly said [2]:

"*Imagination is more important than knowledge*."

8.2. Interface for the disabled

The critical part of a CAD system for the disabled is the interface: build a connection among people with various forms of disabilities. Some interesting research topics are as follows:

- building an interface based on senses other than seeing, e.g. sound or touch;
- providing feedback other than visual, i.e. how does the blind see a lofted surface of a face prepared for plastic surgery;
- developing tools for data migration, i.e. data used for design by deaf people should be sent to production by people who are disoriented; and
- designing methods for particular disabilities, e.g. tools for the blind or deaf;

8.3. Integration of packages

Once systems are designed for various disabilities, they need to be integrated so that communication can be made possible. Some problems worth looking into:

- mapping various designs, e.g. giving a chance to a blind designer to examine the design of a 'normal' person;
- exchange of ideas, e.g. conceptual design among various disabilities;
- exchange of data, i.e. naturally the data base for the blind is not the same as the one for regular people; and
- exchange of design tools, i.e. the way one group performs design may be useful for another and vice versa.

8.4. Model repository

CAD repositories are becoming common for benchmarking and for data sharing. If people with disabilities become integral parts of the design process, they need special repositories:

- repository for the visually impaired;
- repository for the physically challenged; and
- tools to map elements from the various repositories.

9. Challenge no. 8: internet CAD

The internet has changed our lives forever. Some of the changes are welcome, others are annoying, and many are just downright dangerous. CAD has a great potential of benefiting from the tools offered by the interconnection of computers, and the globally available data bases. Although the surface of the potential is barely scratched, the signs of the proliferation of internet CAD tools are clearly visible. A few potential areas of research are listed below.

9.1. The CAD depot

Stores like Home Depot and Office Depot have become hugely popular (and profitable) across the globe. They offer a one-stop solution to home and office needs, e.g. one can go to Home Depot to buy a kitchen cabinet which gets delivered, installed and maintained. A very similar concept can work for CAD as well, i.e. the CAD Depot may offer the following services:

- CAD components (parts), i.e. there is no need to re-invent (re-design, that is) the wheel all the time, one should be able to buy (IGES or STEP files of) common components along with a detailed description of how to use them;
- CAD know-how, e.g. if one needs a good (proven) method of slicing STL models, he should be able to get it from the Depot;
- CAD contractors, i.e. the Depot could be the middle-man between the users and the developers offering much needed consulting services;
- CAD education, i.e. services offered may include on-line courses, school directories, or a major web portal to interface with the web sites and services of major universities world-wide; and
- CAD publishing, i.e. if one has an idea, the Depot may suggest possible journals, conferences and publishers.

9.2. Collaborative CAD

One of the very powerful features of the internet is that it enables the designers who are located far apart to collaborate on a design project. This may sound like an obvious idea, however, there are a number of wrinkles to be smoothed out before this kind of design becomes commonly used:

- data base sharing is doable although CAD data bases tend to be enormous which makes them somewhat difficult to share;
- knowledge base (a not well documented entity especially in the commercial world) sharing is desirable to bring the team to a common level of understanding;
- real-time update can be a challenge especially if the design is complex;
- design platform synchronization becomes a crucial issue to keep up with the various versions of the same design; and
- a common way of thinking (or at least being on a similar wavelength) is required to accomplish a complicated task with various engineers who may come from different backgrounds, speak different languages and have cultural differences.

9.3. Languages and notation

If several groups of engineers want to work together, there has to be an understanding on languages and notation. By this I mean design language and notation:

- there should be a common design language, e.g. something similar to the Booch notation used in object-oriented design;
- there should be a common development language;
- finding a notation for a multi-platform design is an absolute must so that all parties involved can understand each other's design; and
- developing a notation for archival is useful, although this one can be the design notation with some extensions.

9.4. Security

One of the biggest bottlenecks of the internet is security. Just having a simple e-mail account can invade one's sense of security and privacy. Internet CAD is a major target for attacks, e.g. data bases for military equipments could be stolen with a push of the button. Security in internet CAD is *the* most important issue, if it cannot be ensured, the field may not get off the ground at all. Some relevant topics are as follows:

- virus and bug migration from one system to the other;
- data protection against unwanted access as well as internal corruption;
- legal matters related to who may have access to what part of the entire project; and
- issues of jurisdiction, e.g. who is responsible for what part, who owns what, which law can be enforced, what limitations partners have to participate, etc.

10. Challenge no. 9: CAD education

Activities in CAD education seem to lag behind those of the basic sciences like mathematics, physics and biology. The lack of sufficient number of textbooks and an accredited curriculum are examples for the need for more focused work in this area.

10.1. Comprehensive textbooks

If CAD is to mature to the level of physics or computer science, a suite of comprehensive textbooks and teaching aids would be an absolute must. More precisely:

- CAD needs general purpose texts, like the Foley-van Dam of computer graphics, to serve basic undergraduate courses;
- specialized texts, e.g. books on rapid prototyping or on collaborative design, are needed to be used in upper level undergraduate or in graduate courses; and
- teaching aids in the forms of presentation material and software would be most helpful.

The world of academics seems to want to move away from teaching in the direction of pure research. What once was a privilege, i.e. to be allowed to lecture used to be a privilege and was reserved for the most senior professor, is now a punishment. In addition, writing textbooks counts as *one* publication on the resume and hence not many people are willing to spend four years on a book to get a bad annual evaluation in return.

10.2. Accredited curriculum

An important process of globalization is accreditation, i.e. to provide a fairly uniform curriculum across a nation or a region. This way graduates with degrees from various locations can migrate freely because their education will be recognized. To elevate CAD to the level of other accredited disciplines, such as computer science, a three-step procedure should be followed:

- setup of an accreditation agency, at the national level at least, but preferably at the international level;
- establishment of accreditation criteria and procedures, done by the officers of the agency and approved by the community at large; and
- design of an accredited curriculum based on the criteria and guidelines set by the agency.

As in the case of other basic sciences, the accredited curriculum does not have to be bible-like. It can, and should, be flexible enough to give the instructor sufficient freedom to mould the class to his own taste.

10.3. CAD society

The accreditation could be supported, or even driven, by an international society of CAD. This society could be established as:

- a new organization; or
- the merger of many small units.

Currently there exist a number of societies around the world that have members from the CAD community. These members could be asked to form and join a new society, CAD's very own, merging all CAD people into one comprehensive unit with local chapters in each country that wants to participate.

10.4. Journal on CAD education

If CAD education is to be taken seriously, a journal on CAD education would be most helpful. To make this happen, CAD people must address the following issues:

- first, and foremost, research in CAD education must be recognized by the community and by the administration as well;
- second, current CAD journals must recognize CAD education as a new discipline and must publish papers in this field;
- third, a new journal on CAD education should be established; and
- fourth, the CAD community must support the education research with active work, publication and a dedicated conference.

If the current trend continues for another decade or so, all universities turn into 'research universities' where teaching and teaching related activities are considered secondary and burdensome. As a result, the new generation of scientists will be ill-educated incapable of following the footsteps of their mentors. "*Every course that is taught in a substandard manner is one more nail in the coffin of scientific advancements of the next generation*!"

11. Challenge no. 10: CAD theory

Although CAD is a very practical field, the time has come, after some sixty years of existence, to lay the foundation of the field by theories and solid principles. Such a theoretical foundation is useful for many reasons, e.g. distilling basic principles, finding commonalities, etc. Once the basic theorems are identified, the field can be rebuilt theoretically and a comparison with the practical development can be made. At this point theory and practice can complement each other and the two can run parallel, underpinning all developments from both sides of research: theory and practice.

11.1. Axiomatic development

The first thing to do in the theoretical development is to establish a set of axioms, a set of truths accepted as the result of decade long experimentation. That is:

- an axiomatic foundation is desired with a minimum set of axioms; and
- direct consequences are established.

11.2. Basic CAD theories

Then some basic theories need to be established, sitting on top of the axioms, that in turn will form a theory tree, or theory graph.

11.3. Methods of proof

The theories need methods of proof, such as direct or indirect inferences or proof by contradiction. However, there may be many statements that cannot simply be proven by traditional methods, so new methods may be required or accepted, e.g. proof by experiments.

11.4. New directions

One of the advantages of theoretical re-building is that one gains a deeper insight into the field. This insight broadens the horizon and opens up new directions to solve old (and new) problems. Some activities that might be useful are:

- feasibility studies, i.e. theoretical studies could be made before experimentation begins;
- theoretical predictions could be made to guide research and to avoid running into dead ends too often; and
- discoveries are quite likely based on theories and their consequences, e.g. Einstein's very own relativity theory.

12. Conclusions

This paper identified a number of research topics in the general area of CAD. The listed challenges are highly subjective and are the result of the author's experience in the past two decades. The author believes that research in CAD moves way to fast, leaving enormous gaps behind. It was noticed over and over again that published methods, when implemented, did not produce the promised results and that completely new methods had to be designed to solve the problem to suit commercial requirements. It appears that academia and industry have different requirements for research, and until the two can come to a common denominator, a lot of effort will be wasted.

To close this paper, I would like to quote Herbert B. Voelcker, one of the pioneers of CAD [3]. On research this is what he had to say:

> “*It is important to do theoretical research and experimental system building. They are synergistic, and the exclusive pursuit of either can lead to sterile theory or quirky, opaque systems*.”

Education was valued as follows:

> “*Teaching is a marvelous spur to codification, and codification imposes discipline on research. Succinctly, if you can’t teach it with some degree of elegance to able students, you probably don’t understand it*.”

Acknowledgements

The author is indebted to Professors Weiyin Ma and Tony Woo, and to Dr Wayne Tiller for their many valuable comments on the first draft of this paper.

References

[1] Hoffmann B. Albert Einstein: creator and rebel. New York: Penguin Books; 1973.

[2] Mayer J, Holms JP. Bite-Size Einstein: quotations on just about everything from the greatest mind of the twentieth century. New York: St Martin’s Press; 1996.

[3] Voelcker HB, Requicha AAG. Research in Solid Modeling at the University of Rochester: 1972–87. In: Piegl LA, editor. Fundamental Developments of Computer-Aided Geometric Modeling. San Diego, CA: Academic Press; 1993. p. 203–54.

Les A. Piegl is professor in the Department of Computer Science and Engineering, University of South Florida, Tampa, Florida, USA. His research interests are in CAD/CAM, geometric modeling, computer graphics and software engineering. He spent many years researching and implementing geometric algorithms in academia as well as in industry. He is the co-author of the text book The NURBS Book, published by Springer-Verlag. He serves as General Chair of CAD’xx, the annual CAD Conferences and Exhibitions.