

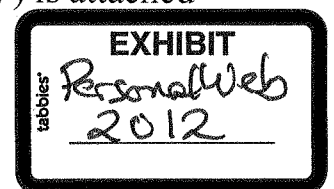
**BEFORE THE PATENT TRIAL AND APPEAL BOARD IN THE UNITED
STATES PATENT AND TRADEMARK OFFICE**

Trial Nos.: IPR 2014-00066
Re: U.S. Patent No. 6,928,442
Patent Owners: PersonalWeb Technologies, LLC & Level 3 Communications
Petitioner: Rackspace US, Inc. & Rackspace Hosting, Inc.

* * * * *

DECLARATION OF ROBERT B. K. DEWAR

1. My name is Robert B. K. Dewar. I reside at 1591 Carpenter Hill Road, Bennington, VT 05201, and am working for the software company Ada Core Technologies in New York, New York, that I founded. I have been retained by PersonalWeb Technologies, LLC. I understand that PersonalWeb is an owner of patents involved in several *Inter Parte Reviews* (IPRs). This declaration relates to U.S. Patent No. 6,928,442 (the '442 patent), and the IPR that it is involved in which I understand to be IPR No. 2014-00066.
2. I am currently Professor Emeritus at New York University in the Department of Computer Science. My curriculum vitae (CV) is attached



at the rear of this declaration. My educational and professional background is summarized below.

3. To summarize my educational and academic background, I was educated at a preparatory school in England, and attended one year of public school there. I then immigrated to the United States where I attended the University of Chicago Lab School. I then attended the University of Chicago where I earned BS and PhD degrees in the field of Chemistry. My PhD work in Chemistry involved substantial use of computers. After receiving my PhD degree, I took a position as Assistant Professor of Computer Science at the Illinois Institute of Technology in 1968.
4. I thereafter received tenure as an Associate Professor of Computer Science at the Illinois Institute of Technology. I was Full Professor of Computer Science, CIMS, NYU, since 1975, and am currently Professor Emeritus at New York University in the Department of Computer Science. I was Chairman of the Department of Computer Science from 1977-1980 at CIMS, NYU. I was Director of Undergraduate Studies, Computer Science from 1981-1987, CIMS, NYU. I was Associate Director of Courant Institute, from 1994-1997.
5. My teaching experience as a Professor includes (i) undergraduate courses relating to Introduction to programming for non-majors, first course for

majors (CS1/CS2 equivalents, taught in Fortran, Pascal, SNOBOL, Ada 95), Assembly language and Computer Architecture, Operating Systems, Microprocessors, Text processing systems, Programming Languages, Formal language theory, Numerical Analysis, Data processing, and Introduction to Algol-68, and (ii) graduate courses relating to Formal language theory, Theory of Computation, Algorithms, Programming Languages, Compilers, Data Communications, Microprocessor Architectures, Real Time Systems, Information Theory, and Programming by Transformation.

6. I have authored or co-authored numerous articles and books relating to computer science, programming, and data processing systems, as shown in my CV.
7. My research has centered on compilers, programming languages, operating systems, and microprocessor architectures. I have engaged in extensive consulting (apart from serving as an expert witness) in the area of operating systems and distributed systems, including writing several operating systems for Honeywell used in distributed applications such as airline reservations and freight management.
8. In 1994 I founded Ada Core Technologies to commercially exploit the work on Ada (programming language) that I was involved with at New

York University. I became CEO and President of Ada Core Technologies.

9. I retired from New York University in 2005.
10. I have received numerous awards including the SIGAda (Special Interest Group on Ada) award for outstanding Ada Community Contributions, as well as several other awards, which are listed in my attached CV.
11. I am familiar with USENETs and Bulletin Board Systems, and used the same in the 1980s and 1990s.
12. I have served as an expert witness on a number of occasions in cases involving patent and copyright issues. A list of several cases in which I served as an expert witness can be found in my attached CV.
13. I am being compensated for my time at the rate of \$500/hour for service that I am providing in connection with these Inter Partes Reviews (IPRs). My compensation is not contingent upon my performance, the outcome of this review, or any issues involved or related to this review.
14. I have reviewed and am familiar with at least the following documents, and any other document mentioned herein: the '442 patent and the prosecution history thereof; U.S. Patent No. 5,649,196 to Woodhill (Woodhill); U.S. Patent No. 4,845,715 to Francisco; U.S. Patent No. 6,135,646 to Kahn, Langer; the Mercer declaration; and the Mercer

deposition transcript. It is my understanding that the '442 patent has an effective filing date of April 11, 1995.

My Understanding of the Law Regarding Patent Validity

15. The following is what I have been told about the law regarding validity of a patent, and it represents my understanding of the same. It is my understanding that Rackspace (who I understand is a “petitioner” in this proceeding) has the burden of proving invalidity in an IPR proceeding by a preponderance of the evidence. As explained below, it is my opinion that the challenged claims of the '442 patent are not invalid under this standard.
16. Anticipation. It is my understanding that a claim of a patent is “anticipated” only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. If even a single element is missing, the claim is not anticipated. It is my understanding that a feature is “inherent” in a reference only if that feature is necessarily present in the reference - not merely probably or possibly present. Furthermore, it is my understanding that in order to anticipate, a prior art reference must not only disclose all elements of the claim, but must also disclose those elements arranged as in the claim.

17. Obviousness. It is my understanding that a patent claim is invalid for “obviousness” if the claimed invention as a whole would have been obvious to one of ordinary skill in the art at the time of the invention in view of a single prior art reference or a combination of prior art references. I understand that a determination of whether a claimed invention would have been obvious requires taking into consideration factors which include assessing the scope and content of the prior art, the differences between the prior art and the claimed invention, and the level of ordinary skill in the art. It is my understanding that the level of ordinary skill in the art is determined by considering the type of problems encountered in the art, the prior art solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology involved, and the educational level of active workers in the field. It is my understanding that a prior art reference may be considered to teach away from the invention when a person of ordinary skill, upon reading the reference, would be led in a direction divergent from the path that was taken by the claimed invention; and that the general rule is that references that teach away cannot serve to create a prima facie case of obviousness.

Level of Ordinary Skill in the Art

18. As of 1995, it is my opinion that a person of ordinary skill in the art to which the '442 patent pertains would likely be a person with a bachelors degree in computer science with ten to fifteen years of teaching or work experience in the field. Of course, one of ordinary skill in the art might have either more formal education with less experience, or less formal education with more experience.

Issues Involving the '442 Patent

19. It is my understanding that the Board, in a decision dated April 15, 2014, instituted a trial regarding the '442 patent for only the following: (a) whether claims 1, 2, 4, 23, 27 and 30 are unpatentable as obvious under 35 U.S.C. §103(a) over Woodhill (Ex. 1004 – U.S. Patent No. 5,649,196) and Francisco (Ex. 1005); (b) whether claim 7 is unpatentable as obvious under 35 U.S.C. §103(a) over Woodhill and Kahn (Ex. 1006 - U.S. Patent No. 6,135,646); and (c) whether claim 28 is unpatentable under §103(a) over Woodhill, Francisco, and Langer (Ex. 1007).

Claim Constructions

20. I have used the following interpretations of claim terms in my declaration. I have been asked to use these constructions for purposes of this proceeding.

21. *Preamble of claim 7 is limiting.* I have been told to assume that the preamble of claim 7 of the ‘442 patent is entitled to patentable weight in this proceeding.

22. *“file” and “data file” (claims 1, 2, 4, 7, 23, and 30).* The ordinary meaning of “file” in this art at the time of the invention was the following definition given in the 1994 Microsoft Computer Dictionary: **“file** A complete, named collection of information, such as a program, a set of data used by a program, or a user-created document. A file is the basic unit of storage that enables a computer to distinguish one set of information from another. A file might or might not be stored in human-readable form, but it is still the ‘glue’ that binds a conglomeration of instructions, numbers, words or images into a coherent unit that a user can retrieve, change, delete, save, or send to an output device.”

(Microsoft Press Computer Dictionary, Second Edition (1994) – Ex. 2011.) This is consistent with a “named” data item referred to at col. 5:48-49 of the ‘442 patent. A “named” data item as described in the ‘442 patent means that the data item is a coherent unit that is individually manipulatable by a user so that a user can selectively retrieve, change, delete, save, or send it to an output device. Thus, one of skill in the art would have understood that a “named” data item is one that is necessarily

a complete coherent unit that a user can selectively retrieve, change, delete, save, or send to an output device. Thus, the ordinary meaning of “file” as described in the 1994 Microsoft Computer Dictionary is both proper and consistent with the description of “file” in the ‘442 patent.

23. A mere *segment* of a basic file is NOT a “file.” Both the specification of the ‘442 patent and the 1994 Microsoft Computer Dictionary confirm this. The specification of the ‘442 patent *differentiates* between a “file” and a mere “segment” of a file, making clear that they are different and that a mere “segment” is not a “file.” As illustrated in Fig. 2 of the ‘442 patent, above, the ‘442 patent differentiates between a “*file*” 120 and a mere “*segment*” 122 of a basic file. (‘442 patent, Fig. 2; col 5:25-35.) The specification of the ‘442 patent explains, as illustrated in Fig. 2, that a “file” is at a different position in the hierarchy than a mere “segment” of a file. (‘442 patent, col. 5:25-30.) The ‘442 patent also explains that while a “file” can be named by a user, a mere “segment” of a file cannot be so named. (‘442 patent, col. 5:44-45; col. 8:25.) Regardless of how “file” is construed, the specification (which includes the drawings) of the ‘442 patent makes clear that a mere “segment” cannot be a “file.”
24. Woodhill’s binary objects and granules are not “files” – they are mere segments of basic files. According to the 1994 Microsoft Computer

Dictionary, a mere segment of a file cannot be a “file” at least because:

(a) a mere segment of a file is not a “complete” collection of information,
(b) a mere segment of a file is not a coherent unit that a user can retrieve,
change, delete, save or send to an output device, (c) a segment typically
has no file header, and (d) a mere segment of a file is not named.

(Microsoft Press Computer Dictionary, Second Edition (1994) – Ex.
2011.)

25. I note that this construction would NOT “preclude a data file from
including smaller data files” in all cases. For example, a *package* such as
a ZIP file may include a plurality of inner files – these would ALL be
“files” under this construction because each of these files would satisfy
the limitations of a “file” discussed above. Each inner file would be
individually named and each would satisfy the definition of a “file” in the
1994 Microsoft Computer Dictionary discussed above. Both before
formation of the ZIP file and after unpacking the ZIP file, each of the
inner files thereof would have a file header and each would be
individually manipulable by users, with each being a coherent unit that a
user can retrieve, change, delete, save or send to an output device. In
contrast, that is *not* the case with the binary objects and granules in
Woodhill. Binary objects and granules in Woodhill do not have headers,

are not individually manipulable as files by users, are not inner files of a package file (e.g., ZIP file), and do not satisfy the definition of a “file” in the 1994 Microsoft Computer Dictionary. A binary object or granule in Woodhill: (a) is determined solely based on size (number of bits) and is NOT based on the extent of any coherent unit that is named or individually manipulable; (b) has no header because it is based solely on size, (c) is NOT a coherent unit that a user can retrieve, change, delete, save or send to an output device, (d) is NOT a “complete” collection of information, and (e) is not named. The binary objects and granules in Woodhill, for example, are mere portions of a basic file, do not have headers, cannot be named, and are not complete coherent units that a user can selectively change, delete, save or send to an output device.

26. One of ordinary skill in the art at the time of the invention would also have understood that a “file” has a file header. The 1994 Microsoft Computer Dictionary explains the ordinary meaning that “*a file header identifies a data file by name, size, and time and date of creation or revision*”; it might also identify the program with which the file was created.” (Microsoft Press Computer Dictionary, Second Edition (1994) – Ex. 2011.) Accordingly, a “file” at the time of the invention, and in the ‘442 patent, must also be construed as including a file header. In contrast,

binary objects and granules in Woodhill do not have file headers, and thus cannot be “files” or “data files” for this additional reason.

27. The proper construction for a “data file” is the same as the construction for a “file” discussed above, except that a “data file” includes data.

28. ***“determining, using at least the name, whether a copy of the data file is present on a particular one of said computers” (claim 2).*** The ordinary meaning of “determine” is “to decide or settle conclusively and authoritatively.” *The American Heritage Dictionary* (1975). (Ex. 2020.)

And the ordinary meaning of “whether” is “if it is so that; if the case is that.” *The American Heritage Dictionary* (1975). (Ex. 2020.)

Accordingly, this clause in claim 2 requires *deciding or settling conclusively and authoritatively, using at least the name, if it is so that a copy of the data file is present on a particular one of said computers.*

This is the proper construction for claim 2.

29. ***“determine whether unauthorized or unlicensed copies of some of the plurality of data files are present on a particular computer” (claim 23).***

As mentioned above, the ordinary meaning of “determine” is “to decide or settle conclusively and authoritatively”, and the ordinary meaning of “whether” is “if it is so that; if the case is that.” *The American Heritage Dictionary* (1975). (Ex. 2020.) Accordingly, this clause in claim 23

requires *decide or settle conclusively and authoritatively [using the list of file names]* if it is so that unauthorized or unlicensed copies of some of the plurality of data files are present on a particular computer. In order to be capable of *determining whether* a particular file, binary object, or granule is present on a given computer, there must be both (a) a capability to determine that it is present on that computer if that is the case, and (b) a capability to determine that it is not present on that computer if that is the case.

30. ***“file name” & “name for a data file” (claims 1, 2, 7, 23).*** A “file name” and “name for a data file” in this art are more than simply an identifier. It cannot reasonably be said that all identifiers are file names. For example, while a fingerprint is an identifier for a person, it is not a “name” for that person. Something is not a “file name” simply because it identifies a file. Any attempt to simply replace “file name” with “file identifier” (or “name” with “identifier”) in a construction would be incorrect and contrary to the claim language itself. Even Woodhill contrasts the difference between a file “name” and a binary object “identifier” (*see e.g.*, file name 40 and file name 80 in Figs. 3-4 of Woodhill, compared to binary object identifier 74 in Fig. 3 of Woodhill), evidencing that they are not necessarily the same thing in this art. One of

ordinary skill in the art at the time of the invention would have understood that there can be a significant difference between a “file name” (or “name for a data file”) and a mere “identifier.” “Name” should not be simply replaced with “identifier.” One of ordinary skill in the art would have recognized that a “file name” (or “name for a data file”) is used to both (a) identify a particular file that the name can be used to differentiate the file from other files, and (b) refer to that file, access that file, search for that file, and address that file so that a user can use the name to selectively retrieve, change, delete, save or send the file to an output device. (E.g., ‘442 patent, col. 16:13-16; col. 17:15-50; col. 19:1-27; col. 23:61-67; col. 24:56-67; col. 34:35-49; and Ex. 2011.) In contrast, identifiers relied upon in the cited art, such as Woodhill’s binary object identifiers 74 and “contents identifiers,” are not “file names” or “names for data files.”

The Challenged Claims Would Not Have Been Obvious

31. It is my opinion that the challenged claims of the ‘442 patent would not have been obvious for at least the following reasons.

Woodhill and Francisco both fail to disclose using a list of content-based file names to determine whether unauthorized

or unlicensed copies of some files are present on a particular computer as in claim 23.

32. Claim 23 of the '442 patent requires: *“using at least said list [of content-based file names] to determine whether unauthorized or unlicensed copies of some of the plurality of data files are present on a particular computer.”* Both Woodhill and Francisco fail to disclose this subject matter. Thus, even the alleged combination (which I do not agree with in any event) fails to meet the claimed subject matter of claim 23.

33. Woodhill certainly does not disclose this subject matter, and it is my understanding that Francisco is relied upon by petitioner in this respect. Indeed, Woodhill fails to disclose deciding or settling conclusively and authoritatively, using a list of binary object identifiers or contents identifiers (alleged file names), if it is so that unauthorized or unlicensed copies of a plurality of data files are present on a particular computer.¹

¹ With respect to Woodhill's granularization restore procedure for example, Woodhill is only analyzing whether a given granule is present in one particular large database file. Woodhill does not and cannot figure out if that granule is present in any of the other files on the local computer 20 or any other computer. The same applies to binary objects in Woodhill.

34. It is my understanding that petitioner thus argues that implementing Francisco's authorization check mechanism in Woodhill's distribution computer network system would have been an obvious improvement. However, Francisco also fails to disclose this claimed subject matter. Therefore, even the alleged combination fails to meet claim 23 for at least the following reasons.

35. **First**, petitioner's allegation on page 31 of the Petition that "Francisco discloses that a copy of a program residing on a computer being requested *can be determined to be unauthorized if there have been any modifications or changes in the software program*" is incorrect.

Francisco's *authenticity* check at comparator 36 determines whether there have been any changes to the program – this is not an authorization check. Francisco's *authorization* check is performed later at comparator 40 *after* the authenticity check, and does not use a list of content based names.

Francisco makes clear that determining whether changes have been made in Francisco is *not* an "authorization" check – it is an "authenticity check.

36. Francisco first performs an "authentication" check at comparator 36 to determine whether the requested program has changed relative to the base version. Then, "subsequent" to the authentication check, Francisco determines at comparator 40 whether the requesting user is authorized to

access the program. (Francisco at Fig. 2; col. 1:40-42; and col. 2:65 to col. 3:35.) The program identifier is compared with only a single value in comparator 36 during the “*authentication*” check, and is not compared with anything in comparator 40 during the “*authorization*” check.

37. Francisco’s *authentication* check at comparator 36 merely compares a program identifier (electronic identification indicia) 34 from generator 32 with a single program identifier 12. (Francisco at Fig. 2; col. 1:40-42; col. 2:65 to col. 3:20; and col. 4:12-28; and Mercer Dep. 60 [Ex. 2015].) Thus, there is no comparison of an identifier (12 or 34) to a list of content-based names in the authentication check at comparator 36. Moreover, comparator 36 merely performs an authentication check to make sure that the requested program has not changed – this is *not* an authorization check and does *not* determine whether the requested program is unauthorized or unlicensed.

38. If there is a match between identifiers 12 and 34 in comparator 36, then the requested program has not changed and it is only then that Francisco moves on to comparator 40 for determining whether the requesting user is authorized to access the program. (Francisco at Fig. 2; col. 3:17-36; and col. 4:30-51.) Francisco explains that the *authorization* check is performed “subsequent” to the *authentication* check. (Francisco at Fig.

2; and col. 1:40-42.) Because the authorization check at comparator 40 is “subsequent” to the authentication check, they are not the same and are different checks. In making the authorization check at comparator 40, Francisco compares an identification indicia of the requesting user with a user profile 42. (Francisco at Fig. 2; col. 3:20-36; col. 4:30-33; and col. 4:40-51.) Comparator 40 compares these two user indicia with each other in order to determine whether access is authorized. *Id.* See also Francisco at col. 4:48-51. Francisco’s comparator 40 does *not* compare program identifier 34 with anything – it merely forwards identifier 34 in a “program release” signal if the two user indicia match each other. No list of content-based file names is used at comparator 40 for any reason.

39. Accordingly, Francisco compares the program identifier (12 or 34) with a single value in comparator 36 to make sure that the file is *authentic* (*i.e.*, that it has not changed), and if the file is found to be authentic then Francisco later compares *user information* in another comparator 40 to determine if access is authorized. Thus, Francisco’s program identifier 34 (or 12): (i) is only compared to a single value for determining whether the program has changed, and (ii) is not compared to anything for determining whether access by a particular user is authorized. No list

of content-based file names is used during Francisco's authorization check at comparator 40.

40. The only possible alleged "list" of content-based file names in Francisco could be argued to be in the "electronic identification inicia library" in the upper right corner of Fig. 2 of Francisco. However, *nothing from this library is used during Francisco's authorization check at comparator 40.* Accordingly, even providing Francisco's system in Woodhill fails to meet the subject matter of claim 23. If Francisco's system were provided in Woodhill (with Woodhill's binary object identifiers 74 or records 58 possibly taking the place of Francisco's program identifiers), the invention of claim 23 still would not be met because the combination would NOT use a list of content-based file names to determine whether unauthorized or unlicensed copies of a plurality of data files are present on a particular computer.

41. **Second**, as explained above in the claim construction section, claim 23 requires deciding or settling conclusively and authoritatively, using a list of content-based file names, if it is so that unauthorized or unlicensed copies of a plurality of data files are present on a particular computer. It is my understanding that petitioner relies upon Francisco for this subject matter. In order to be capable of *determining whether* a particular file,

binary object, or granule is present on a given computer, there must be both (a) a capability to determine that it is present on that computer if that is the case, and (b) a capability to determine that it is NOT PRESENT on that computer if that is the case. Woodhill and Francisco are both incapable of at least the latter, because for a given file, binary object or granule they are only capable of analyzing information relating to *one file* on a given computer. They are blind regarding other files on that computer and elsewhere in the system. In Francisco (relied upon in this respect by petitioner), a single program (presumably on some server) is “selected” and requested to be released. (Francisco at col. 2:66-69 and Fig. 2.) Francisco then generates an identification signal 34 for that particular program and sends it to comparator 36. (Francisco at col. 2:66 to col. 3:3, and Fig. 2.) *Francisco does not analyze or generate any identification signal for any other programs on the server on which the selected program is presumably located.* Therefore, Francisco cannot determine whether a duplicate of that program, or a different and unlicensed version of that program, is located at a different location on that server. Because Francisco’s Fig. 2 procedure is blind to all programs on a given computer except for the one that is selected, Francisco is incapable of determining that a given program is *not present* on that

particular server if that is indeed the case. Unlike the '442 patent, Francisco (and Woodhill) does not search a list of content-based names for determining whether unauthorized or unlicensed files are present at a particular computer.

42. Assume a situation in Francisco where two versions of a given program are on a given server, where version A is licensed but version B is unlicensed. If a user selects version A of the program, then as explained above Francisco would do no analysis of version B and is incapable of figuring out that unlicensed version B is present on the server. Again, Francisco does not use a list of content-based names to determine whether unlicensed or unauthorized programs are on a particular server/computer.

One skilled in the art would not have modified Woodhill to add Francisco's or Kahn's access control system because Woodhill already has access control that is unrelated to binary object identifiers.

43. Given that Woodhill already has access control, there would have been no logical reason to have modified Woodhill to have provided further access control with Francisco or Kahn. One skill in the art, as a matter of common sense and logic which a person of ordinary skill in the art is presumed to have, would not think to layer additional and unnecessary

access control onto the existing access control of Woodhill. Thus, there would have been no motivation or logical reason for the alleged modification. Claims 1, 2, 4, 7, 23, 27, 28 and 30 define over the cited art, and are nonobvious, in this respect.

44. During the backup procedure, Woodhill separates each file being backed up into multiple data streams. (Woodhill, col. 7:41-43.) “These data streams may represent regular data, extended attribute data, *access control list data*, etc.” (Woodhill, col. 7:44-47.) Petitioner’s expert acknowledges that this “access control list data” relates to authorization checking in Woodhill. (Mercer Dep. 113, 115 [Ex. 2015].) Referring to Fig. 5A of Woodhill, it can be seen that this “access control list data” exists in step 132 *before* binary object identifiers 74 are even calculated in step 138. (Woodhill, Fig. 5A, col. 7:40-67.) Indeed, binary objects in Woodhill include such extended attribute data and access control list data, and Woodhill creates binary object identifiers 74 in part by hashing such extended attribute data and access control list data. (Woodhill, col. 7:41-67 and Fig. 5A.) Given that Woodhill’s “access control list data” exists *prior* to binary object identifiers 74 being created, it would be illogical and nonsensical to modify Woodhill as alleged by petitioner to use the binary object identifiers 74 for checking authorization/unauthorization.

Moreover, as Woodhill already had an authorization/deauthorization system based on this “access control list data” (which is unrelated to binary object identifiers 74), there would have been no logical reason to have modified Woodhill as alleged by petitioner to add another authorization/unauthorization checking feature to Woodhill.

One would not have modified Woodhill to check whether access to a file by a computer was authorized when the computer already had the current version of that file.

45. In Woodhill as allegedly modified by Francisco or Kahn, petitioner contends that the “request” of claims 1 and 7 is the “update request” described in Woodhill at column 17:42-46. In Woodhill’s restore procedure, the local computer 20 *already has* the most recent (current) version of the large database file at issue. (Woodhill at col. 17:39-40, col. 17:59-60.) Because the local computer 20 already has the most recent (current) version of the large database file at issue, there is no logical reason to have modified Woodhill to implement a system for checking whether that same local computer 20 is authorized to access a previous version of that same file. One would have assumed that the local computer 20 is permitted to access a prior version of a file if that computer already has the current version of that file. Given that the local computer 20 already has the most recent version of the file, as a matter of

common sense there would have been no logical reason to have checked whether that same computer is authorized to access the previous version of that same file.

46. Petitioner's alleged reason for making the alleged modification is also fundamentally flawed. It is my understanding that petitioner's rationale for the alleged modification is "to only provide copies of the requested file to an authorized party identified in a table as set forth in Francisco and not provide copies of files to unauthorized parties as set forth in Francisco." However, the local computer in Woodhill's granularization restore procedure already has the most recent version of this file, so there would be no concern of that computer/user not having authorization. Moreover, Woodhill already has an access control system. (Woodhill, col. 7:44-47.) If Woodhill already has an access control system, there would have been no logical reason to have added another one. Indeed, the fact that Woodhill already had an access control system would NOT have motivated one of ordinary skill in the art to have added another one to Woodhill.

47. Woodhill also describes a "self-audit" procedure at col. 18:10-38. There also would have been no logical reason to have modified that procedure. Woodhill's auditing procedure is for performing "self-audits" of the

system. (Woodhill at col. 18:10-12.) Given that the system already has the binary object, as a matter of common sense there would have been no logical reason in a “self” auditing system to have checked whether the system is authorized to access that binary object which it already has. Moreover, there are no potential unauthorized users involved in Woodhill’s self-audit procedure. Still further, there is no “request” in the self-auditing procedure that includes a binary object identifier 74 or a contents identifier – so claims 1 and 7 would not be met in any event.

Francisco and Woodhill teach away from claim 23.

48. It is my understanding that a prior art reference may be considered to *teach away* when a person of ordinary skill, upon reading the reference, would be led in a direction divergent from the path that was taken by the patent. Here, Francisco and Woodhill both would have led one of ordinary skill in the art away from the path taken by the inventors. While both Woodhill and Francisco have content-based identifiers and both have access control systems, neither Woodhill nor Francisco compares their content-based identifiers to anything for determining whether access is authorized or for searching for unauthorized copies. And neither Woodhill nor Francisco uses a list of content-based names for

determining whether unauthorized or unlicensed copies of files exist on a particular computer.

49. Francisco's identification "indicia" 12 and 34 are content-based.

(Francisco at Fig. 2; col. 2:29-42; and col. 3:49 to col. 4:17.) Francisco's *authentication* check at comparator 36 compares a program identifier (electronic identification indicia) 34 from generator 32 with a single program identifier 12. (Francisco at Fig. 2; col. 1:40-42; col. 2:65 to col. 3:20; and col. 4:12-28.) Comparator 36 merely performs an authentication check to make sure that the requested program has not changed – this is not a determination of whether a particular user is authorized to access the program. If there is a match between 12 and 34 in comparator 36, then the requested program has not changed and it is only then that Francisco moves on to comparator 40 for determining whether the requesting user is authorized to access the program.

(Francisco at Fig. 2; col. 3:17-36; and col. 4:30-51.) Francisco clearly explains that the *authorization* check is performed "subsequent" to the *authentication* check. (Francisco at Fig. 2; and col. 1:40-42.) In making the authorization check at comparator 40, Francisco compares an identification indicia of the requesting user with a user profile 42.

(Francisco at Fig. 2; col. 3:20-36; col. 4:30-33; and col. 4:40-51.)

Comparator 40 compares these two user indicia with each other in order to determine whether the requesting user is authorized to access the program. *Id.* See also Francisco at col. 4:48-51. Francisco's comparator 40 does *not* compare program identifier 34 (or 12) with anything in making the authorization check. There is no list of alleged content-based names used in Francisco's authorization check at comparator 40 – contrary to claim 23.

50. Woodhill's binary object identifiers 74 are also content based.

(Woodhill at col. 8:22-31.) But Woodhill does not use the binary object identifiers 74 in its access control system. During the backup procedure, Woodhill separates each file being backed up into multiple data streams. (Woodhill, col. 7:41-43.) "These data streams may represent regular data, extended attribute data, *access control list data*, etc." (Woodhill, col. 7:44-47.) Referring to Fig. 5A of Woodhill, it can be seen that this "access control list data" exists in step 132 before binary object identifiers 74 are even calculated in step 138. (Woodhill, Fig. 5A, col. 7:40-67.) Indeed, binary objects in Woodhill include such extended attribute data and access control list data, and Woodhill creates binary object identifiers 74 in part by hashing such extended attribute data and access control list data. (Woodhill, col. 7:41-67 and Fig. 5A.) Given that

Woodhill's "access control list data" exists *prior* to binary object identifiers 74 even being created, Woodhill cannot use the binary object identifiers 74 in its access control system and teaches away from doing so. Again, there is no list of alleged content-based names used in any authorization check in Woodhill.

51. Therefore, while both Woodhill and Francisco have content-based identifiers and both have access control features, neither Woodhill nor Francisco compares their content-based identifiers to anything for determining whether access is authorized, or for determining whether unauthorized or unlicensed copies of files exist on a particular computer. These references teach away from claim 23.

**Prior art elements are not used for their intended purpose
regarding claim 23.**

52. Woodhill's binary object identifiers 74 and "contents identifiers", and Francisco's identifiers 12 and 34, are NOT used for comparisons to determine whether access is authorized. And, unlike claim 23, there is no list of such content-based identifiers in either Woodhill or Francisco that is used to determine whether unauthorized or unlicensed copies of files exist on a particular computer. Thus, any modification to meet the challenged claims would result in these identifiers NOT performing the same functions that they are known to perform in Woodhill and

Francisco. This is strong evidence of nonobviousness in my opinion. While both Woodhill and Francisco use content-based identifiers for various purposes, neither compares such identifiers to determine whether access to a file is authorized. And neither uses a list of such content-based identifiers to determine whether unauthorized or unlicensed copies of files exist on a particular computer. The inventors of claim 23 conceived of a new and nonobvious use for substantially unique content-based identifiers which is not disclosed or suggested in the relied upon art.

Granules corresponding to “contents identifiers” in the alleged request in Woodhill are never provided to the requesting computer in response to the request (claims 1 and 7).

53. Claims 1 and 7 each require a “request” for a data file, which request includes the name of that particular data file. Claims 1 and 7 further require that, in response to the request, a copy of the requested file is provided from a given one of the computers and (a) only to licensed parties (claim 1), or (b) is not provided to unlicensed or unauthorized parties (claim 7). This subject matter cannot be met if it is argued that “files” are granules and “names” are “contents identifiers” with respect to Woodhill’s granularization restore procedure (regardless of whether it is modified).

54. In the alleged Woodhill combination, with Francisco or Kahn, granules corresponding to “contents identifiers” in the alleged request (“update request”) are NEVER provided from the remote backup server 12 to the requesting computer in response to the “update request.” (Woodhill at col. 17:18-67.) Thus, Woodhill/Francisco and Woodhill/Kahn cannot meet claims 1 and 7 if the claimed data files are argued to be granules and the names are considered “contents identifiers.” Moreover, no name of the large database file at issue is included in the “update request,” so that the claimed data file in claims 1 and 7 cannot be considered to be the large database file.

55. In Woodhill, and in any modification of Woodhill/Francisco or Woodhill/Kahn, granules corresponding to “contents identifiers” in the alleged request (“update request”) are never provided to the requesting computer in response to the update request. (Woodhill at Fig. 5I and col. 17:18-67.) In the relied upon restore procedure, no granule having an identifier in the “update request” is ever provided to the requesting local computer 20 (alleged licensed party). The “contents identifiers” in the alleged “request” were calculated in step 444 and thus are NOT for granules that are being requested – the requesting local computer 20 already has the granules corresponding to the “contents identifiers”

calculated in step 444 and therefore they are never provided to the local computer even in the allegedly modified system. And because granules identified in the request are never sent from remote server 12 to the local computer in the restore procedure (the local computer already has them), it would not make sense to use identifiers for such granules to determine whether the system is authorized to provide such granules to the local computer. Thus, in addition to all other reasons set forth above, such combinations (which I disagree with in any event) fail to meet claims 1 and 7.

56. Turning to the “contents identifiers” for the reconstituted granules in Woodhill, the “contents identifiers” for the granules reconstituted at col. 17:48-50 are only used for comparison in box 450 of Fig. 5i *after* the Fig. 5h reconstitution of those granules, and are *never* included in any “request” regarding a data item.

Binary objects corresponding to Record 58 allegedly in the “update request” are not provided to the requesting computer in response to the “update request” (claims 1 and 7).

57. In the alleged Woodhill/Francisco and Woodhill/Kahn combinations binary objects corresponding to the binary object identification records 58 allegedly in the “update request” are NOT provided to the requesting

computer in response to the update request. (Woodhill at col. 17:18 to col. 18:9.) The granularization restore procedure in Woodhill as allegedly modified by Francisco or Kahn transmits select granules – not binary objects. And record 58 and binary object identifier 74 do not identify particular granules. Thus, Woodhill/Francisco and Woodhill/Kahn cannot meet claims 1 and 7 if, as alleged by petitioner, the claimed data items are binary objects and the identifiers are considered binary object identifiers 74 or binary object identification records 58.

58. As explained above, claims 1 and 7 each require a “request” for a data file, which request includes the name of that particular data file. Claims 1 and 7 further require that, in response to the request, a copy of the requested file is provided from a given one of the computers and (a) only to licensed parties (claim 1), or (b) is not provided to unlicensed or unauthorized parties (claim 7). This subject matter cannot be met if the claimed files are considered to be Woodhill’s binary objects and the identifiers to be Woodhill’s binary object identifiers 74 or records 58 as argued by petitioner, because the binary objects corresponding to the identifier allegedly in the “request” are not disclosed as being transmitted

to the local computer 20 in response to the request in Woodhill's granularization restore procedure.

59. In Woodhill, and in any modification of Woodhill, binary objects corresponding to binary object identification records 58 allegedly in the request ("update request") are *not* provided to the requesting computer in response to the request. (Woodhill at Fig. 5I and col. 17:18-67.)

Woodhill never discloses such an occurrence, and instead indicates to the contrary. The restore procedure transmits only particular granules based on whether they have changed – not binary objects. I also note that the granularization restore procedure in Woodhill is different than the self-audit procedure in that the different procedures transmit different items, use different comparisons, and so forth. The two procedures cannot be combined in any logical manner.

60. Thus, in the granularization process upon which petitioner relies, Woodhill is not designed to provide, and *never* provides, any binary object to the local computer 20 in response to the "update request." No binary object sequence of bits is ever sent from the backup file server 12 to a local computer 20 in the restore procedure of Woodhill's granularization feature (regardless of whether modified by Francisco/Kahn). In Woodhill's granularization process, the backup file

server 12 sporadically sends only select “granules” of a large database file to the local computer. (Woodhill, col. 17:60-65.) Binary object identifiers 74 and records 58 do not identify granules, and are not names for granules. Woodhill’s granularization embodiment does *not* send large database files or binary objects from server 12 to the local computer – just select granules.

**Binary objects and granules are not “data files” as required
by the challenged claims.**

61. The binary objects and granules in Woodhill are not “files” or “data files,” for the reasons explained above in the claim construction section.
62. It is my understanding that petitioner relies on a “granule” or “binary object” from Woodhill as the alleged “data file” with respect to all challenged claims. Because these are not data files as explained above in the claim construction section, petitioner’s allegations regarding all challenged claims are incorrect.
63. Claim 1 for example states as follows, *inter alia*: “obtaining a *name for a data file*, the name being based at least in part on a given function of the data, wherein *the data used by the given function to determine the name comprises the contents of the data file*; and in response to a *request for the [a] data file*, the request including at least the name of the particular file, causing a copy of the file to be provided from a given one of the

plurality of computers . . .” There is no “request” for a “data file” in Woodhill’s granularization restore procedure relied upon by petitioner. Neither a binary object nor a granule is a “data file.” Woodhill’s “update request,” relied upon by petitioner, does not contain a name for a requested “data file” as required by claims 1 and 7.

64. Woodhill’s granularization embodiment relates to *large* database files each of which contains *many* binary objects and *many* granules (Woodhill, col. 14:52-67; col. 15:11-17; col. 17:18-22; col. 17:31-35.) Petitioner relies upon the “update request” described by Woodhill at col. 17:42. However, that “update request” does not request a large database file (the only type of file handled by Woodhill’s granularization embodiment) or any other “data file,” and does not contain any name for a large database file. An alleged request for a binary object is not the same as a request for a large database file. Accordingly, there is not even a “request” for a “data file” in Woodhill’s granularization restore procedure, and there is no such request that includes a name for a requested “data file.” Woodhill’s granularization restore at column 17 sends particular granules (not binary objects, and not an entire large database file) to a local computer. (Woodhill, col. 17:50-64.) Indeed, a purpose of Woodhill’s granularization embodiment is to avoid

transmitting an entire large database file, which teaches away from petitioner's allegations regarding claims 1 and 7. Accordingly, even the alleged combinations fail to meet claims 1 and 7.

65. Furthermore, while Woodhill mentions Binary Object Identification Record 58 at col. 17:44, Woodhill indicates that portions (62 and 72) of the Record 58 other than the binary object identifier 74 are used to identify a given binary object in a message. (Woodhill, col. 11:65 to col. 12:5.) Woodhill indicates in this respect that binary object identifiers 74 are *not* used in any alleged "request" for a binary object. (Woodhill, col. 11:65 to col. 12:5.) Woodhill indicates that binary object identifiers 74 are *not* used for accessing binary objects. *Id.*

66. Petitioner briefly mentions Woodhill's "self-audit" embodiment described at col. 18:10-38. (Ex. 1010, pg. 32.) However, Woodhill's self-audit embodiment also fails to disclose or suggest the subject matter of claims 1 and 7, regardless of whether modified by Francisco or Kahn. (Woodhill, col. 18:10-38.) There is no "request" in Woodhill's self-audit procedure that includes a binary object identifier 74 (let alone a name for a "data file"). Woodhill explains in the description of the self-auditing procedure that the Record 58 is "stored in File Database 25." (Woodhill, col. 18:16-19.) While Woodhill states that the Record 58 is "stored in

File Database 25”, the Record 58 is never described as part of a “request” for the binary object. (Woodhill, col. 18:16-19.) Moreover, Woodhill elsewhere indicates that, in a message, portions of the Record 58 other than the binary object identifier 74 are used to identify a binary object. (Woodhill, col. 11:65 to col. 12:5.) Accordingly, in Woodhill’s self-auditing procedure there is no description of a “request” for a binary object or anything else that includes a binary object identifier 74. To the contrary, Woodhill indicates that any such “request” would not include a binary object identifier 74. (E.g., Woodhill, col. 11:65 to col. 12:5.) Additionally, there is no request for a “data file” in Woodhill’s self-audit procedure as explained above. Woodhill restores a randomly selected binary object – not a “data file.” (Woodhill, col. 18:25.) A binary object is not a “data file.” And neither a binary object identifier 74 nor a binary object identification record 58 is a “name” for a data file. Moreover, because Woodhill’s “self-audit” embodiment is an automatic procedure performed by the system on a periodic basis to check itself, it is entirely unrelated to the final lines of claims 1 and 7 (e.g., “a copy of the requested file is only provided to licensed parties”). Since Woodhill’s “self-audit” embodiment is an automatic procedure for checking itself,

there would have been no logical reason to have implemented an authorization checking system in Woodhill's "self-audit" procedure.

67. Furthermore, it would be incorrect to switch back and forth between Woodhill's granularization embodiment, and Woodhill's "self-audit" embodiment, in an attempt to mix and match to meet all elements of a given claim. These are different embodiments that function separately on different bases. It is my understanding that in order to meet a claim a reference must not only disclose the elements of the claim, but must also disclose those elements "arranged or combined in the same way as in the claim."

Binary objects and granules cannot represent a "digital image" as required by claim 7.

68. Claim 7 requires a "request" for a data file, which request includes the name of that particular data file. Claim 7 further requires that, in response to the request, a copy of the requested file is provided from a given one of the computers. Petitioner relies on the "update request" in Woodhill's granularization restore procedure for the alleged "request," and contends that the granules represent "data files" and that the "contents identifiers" represent names for those alleged "files." (Ex. 1010 at 32-33 and 54.)

69. However, claim 7 expressly recites “the contents of said data file representing a *digital image*.” This cannot occur with a granule (or a binary object) in Woodhill, because a granule (or binary object) in Woodhill is only a small portion of a large database file and cannot itself represent a digital image.
70. Woodhill’s granularization restore procedure which petitioner seeks to modify is used for processing *large database files* each of which includes many binary objects and many granules. (‘442 patent at col. 14:52-65, col. 17:18-35.) In 1995, one of ordinary skill in the art would have expected a “large database file” of the type handled by Woodhill’s granularization procedure to be at least 1 to 2 gigabytes in size. Woodhill explains that a binary object is 1 megabyte in size, and that a granule can be 1 kilobyte. (Woodhill, col. 7:54-55, col. 14:58-67.) Given such sizes, a single 1 gigabyte large database file to be processed by Woodhill’s granularization restore procedure would include at least 1,024 binary objects and 1,048,576 granules. Thus, each granule is a small portion of a binary object, and each binary object is a small portion of the large database file. While a large database file in Woodhill could possibly itself represent a “digital image,” a small segment thereof such as a granule or binary object could not. Each granule and binary object could

possibly represent a small portion of the data for a digital image, but could not itself represent a digital image as required by claim 7.

Accordingly, a granule or binary object in Woodhill cannot be the type of “file” (a “file” that represents a digital image) called for in claim 7.

Accordingly, even the alleged modification to Woodhill fails to meet claim 7 because a granule or binary object cannot represent a “digital image” as called for in claim 7.

71. Additionally, any alleged modification to Woodhill’s self-audit procedure also cannot meet claim 7, because again a binary object (a portion of an overall file in Woodhill) cannot itself represent a “digital image.” Moreover, there is also no “request” for a data file in Woodhill’s self-audit procedure that includes a binary object identifier 74 of a particular binary object. And, as explained above, there are no unlicensed or unauthorized parties involved in Woodhill’s “self” audit procedure, so the claim also cannot be met in this respect and there would have been no logical reason to have provided an authorization check in a “self” audit procedure.

72. Furthermore, no “name” of the large database file is included in Woodhill’s “update request.” Therefore, the claimed data file called for in claim 7 cannot be considered the large database file in Woodhill/Kahn

because no name therefor is provided in any alleged “request for the data file” as required by claim 7.

The alleged identifier in the argued combinations is not a “file name” and is not used for accessing.

73. As explained above in the claim construction section, one of ordinary skill in the art would have recognized that a “file name” and “name for a data file” are used to both (a) identify a particular data item so that the name can be used to differentiate the data item from other data items, and (b) refer to that data item, access that data item, search for that data item, and address that data item so that a user can use the name to selectively retrieve, change, delete, save or send the item to an output device. (E.g., ‘442 patent, col. 16:13-16; col. 17:15-50; col. 19:1-27; col. 23:61-67; col. 24:56-67; col. 34:35-49; and Ex. 2011.)

74. The alleged identifiers in the Woodhill/Francisco and Woodhill/Kahn combinations are NOT “file names” or “names for data files.” Woodhill’s binary object identifiers 74 are not “file names” for binary objects, and Woodhill’s contents identifiers are not “file names” for granules. And Francisco’s program identifiers are also not “file names” for data files. Because these are not “file names” or “names for data files,” even the alleged modification fails to meet at least claims 1, 11, 24 and 32.

75. Woodhill uses “names” for files (see name 40 in Fig. 3 of Woodhill), but does *not* provide names for binary objects or granules. Woodhill merely employs a hash to compare same file components to determine whether there has been a change – Woodhill does not use a hash as a file name, or as a substitute for a name, in a computer network. Indeed, Woodhill’s granularization process upon which petitioner relies relates to a large database file that has many binary objects and many granules – these granules and binary objects of such a large database file are not provided with file names in Woodhill’s granularization process and there would have been no logical reason to have done so.

76. Something is not a “file name” simply because it is an “identifier.” For example, while a fingerprint is an identifier for a person, it is not a “name” for that person. One of ordinary skill in the art at the time of the invention would have understood that there is a significant difference between a “file name” and a mere “identifier” as explained above.

77. Woodhill’s “binary object identifiers 74” are not used to access binary objects, search for binary objects, or address binary objects. Binary object identifiers 74 are not even mentioned in columns 16-17 of Woodhill. When a Binary Object Identification Record 58 is provided such as at col. 17:44, and a binary object is accessed, Woodhill expressly

explains how this is done and it does NOT involve any binary object identifier 74. (Woodhill at col. 11:65 to col. 12:7, and col. 9:18-20.) Thus, Woodhill itself describes that fields 62 and 72 of Binary Object Identification Record 58 are used to access binary objects (i.e., the field for binary object identifier 74 is NOT used). Thus, binary object identifiers 74 are not compared to anything or otherwise used in any respect when accessing, searching for, or addressing binary objects in the allegedly modified system. They are not “file names” or “names for data files.”

78. Likewise, Woodhill’s “contents identifiers” are not used to access granules, search for granules, or address granules. For instance, in Woodhill’s procedure for reconstituting a binary object on a granule-by-granule basis, “contents identifiers” are not used or even mentioned. (Woodhill at Fig. 5H and col. 16:16 to col. 17:17.) This same procedure is used in the restore procedure. (Woodhill at col. 17:46-50.)

79. Therefore, the “file name” or “name for a data file” aspect of the challenged claims is not disclosed by Woodhill as allegedly modified by Francisco or Kahn.

80. Additionally, while Woodhill mentions names in its claims (*e.g.*, Woodhill at col. 22:3-9; col. 22:35-50; col. 22:67 to col. 23:2; col. 23:10-

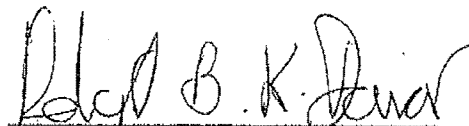
19; and col. 24:24-39), this is not “prior art” to the challenged claims. The “name” language in Woodhill’s issued claims is not “prior art” to the challenged claims. The subject matter added to Woodhill on January 5, 1996 (including the “name” language in Woodhill’s claims) was added after the April 11, 1995 effective filing date of the ‘442 patent, was not in Woodhill’s originally-filed specification, and is not “prior art” to the challenged claims of the ‘442 patent (see prosecution history of Woodhill at Ex. 2007). Woodhill’s binary object identifiers and “contents identifiers” are not names for binary objects or granules. Binary objects and granules in Woodhill are not files and thus do not have file names. One of ordinary skill in the art reviewing Woodhill’s specification as originally filed would have understood that binary object identifiers 74 were not “names” or “file names” for binary objects, and that “contents identifiers” were not “names” or “file names” for granules. Many identifiers are not names. For example, while a fingerprint is an identifier for a person, it is not a “name” for that person. It is my understanding that subject matter added to Woodhill after the effective filing date of the challenged claims is not “prior art” and cannot be used against the challenged claims in this proceeding.

MD4, MD5 and SHA of Claim 28 Would not have been obvious.

81. Claim 28 requires an MD4, MD5 or SHA. It is my understanding that petitioner does not contend that MD4 or SHA would have been obvious to have used in Woodhill or Francisco. However, I understand that petitioner does contend that it would have been obvious to have modified Woodhill/Francisco to use an MD5 based on Langer.
82. It is my opinion that it would *not* have been obvious to have calculated Woodhill's Binary Object Identifier 74 (or "contents identifier") using an MD5 hash function. There were thousands and thousands of hashes known in the art at the time of the invention. There would have been no logical reason to have selected an "MD5", out of the many thousands of possible hashes, for use in Woodhill. An MD5 is a non-reversible cryptographic hash – Woodhill's alleged hashes in column 8 are not. Moreover, an MD5 produces a 16 byte hash value, whereas Woodhill desires a 4 byte hash value. (Woodhill, col. 8:1-3.) Thus, an MD5 would not be compatible with Woodhill's system (whereas many other available hashes would have been), and there would have been no logical reason to have selected it from the large number of possible other hash functions that were available at the time of the invention.

MISCELLANEOUS

83. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested proceeding before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross examination in the proceeding and that cross examination will take place within the United States. If cross examination is required of me, I will appear for cross examination within the United States during the time allotted for cross examination.
84. I reserve the right to supplement my opinions in the future to respond to any new arguments that petitioner may raise and take into account new information as it becomes available to me.
85. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

A handwritten signature in black ink, appearing to read "Robert B.K. Dewar", written over a horizontal line.

Robert B.K. Dewar

Curriculum Vita

Robert B. K. Dewar

Born: June 21, 1945

EDUCATION

| | |
|--|-----------|
| Copthorne Prep school Copthorne, Sussex, England | 1952-1958 |
| Won open scholarship to St. Pauls, 1958 | |
| St. Paul's Public School London, England | 1958-1959 |
| Emigrated with family to USA in 1959, with permanent residence status | |
| University of Chicago Lab School Chicago, IL | 1959-1961 |
| Admitted to University of Chicago as a University Scholar | |
| University of Chicago, the college | 1961-1964 |
| Graduated with BS in chemistry in 1964, having already completed all course work for the graduate program, and having passed the test for admission to PhD research. | |
| University of Chicago, graduate school | 1964-1968 |
| Graduated with PhD in chemistry in 1968 Awarded the Elizabeth Norton prize for excellence in submitted thesis | |

EMPLOYMENT

| | |
|--|------------------------|
| Assistant Professor of Information Science Illinois Institute of Technology, Chicago IL | 1968-1970 |
| Received tenure from IIT, 1970 | |
| Associate Professor of Computer Science Illinois Institute of Technology, Chicago IL | 1970-1975 |
| Visiting Fellow Electronic Computer Laboratory University of Leeds | 1971-1974 (summers) |

| | |
|---|--------------|
| Research Associate Professor of Computer Science Courant Institute of Mathematical Sciences New York University, New York (www.cs.nyu.edu) | 1975-1976 |
| Director of Graduate Studies, Computer Science CIMS, NYU | 1975-1976 |
| Received tenure from NYU, 1976 | |
| Full Professor of Computer Science CIMS, NYU | 1976-2005 |
| Chairman of department of Computer Science CIMS, NYU | 1977-1980 |
| Director of Undergraduate Studies, Computer Science CIMS, NYU | 1981-1987 |
| Associate Director of Courant Institute | 1994-1997 |
| Founded Ada Core Technologies in 1994 | |
| President and CEO Ada Core Technologies (www.gnat.com) | 1994-present |

TEACHING EXPERIENCE

(representative list of course topics taught at Illinois Institute of Technology and New York University)

Undergraduate

Introduction to programming for non-majors
 First course for majors (CS1/CS2 equivalents, taught in Fortran, Pascal, SNOBOL, Ada 95)
 Assembly language and Computer Architecture
 Operating Systems
 Microprocessors
 Text processing systems
 Programming Languages
 Formal language theory
 Numerical Analysis
 Data processing
 Introduction to Algol-68

Graduate

Formal language theory
 Theory of Computation
 Algorithms

Programming Languages
Compilers
Data Communications
Microprocessor Architectures
Real Time Systems
Information Theory
Programming by Transformation

In preparation

Technical Aspects of Software Copyright Issues (with NYU Law School)

PUBLICATIONS

1. Fleischer, E.B., Stone, A.L., Dewar, R.B.K., Wright, J.D., Keller, C.E., and Petit, R., *The Molecular Structure of the Complex of Cyclooctatetraene and Iron Pentacarbonyl*, JASC 88, 3158, 1965.
2. Fleischer, E.B., Kaiser, E.T., Langford, P., Hawkinson, S., Stone, A.L., Dewar, R.B.K., *The Molecular Structure of Five- and Six-Membered Cyclic Sulphonates*, Chem. Comm., 197, 1967.
3. Dewar, R.B.K. and Manacher, G.K., *Some New File-Handling Input-Output and Monitoring Procedures for Use with the SNOBOL Language*, ICR QUARTERLY REPORT 12, University of Chicago, 1967.
4. Dewar, R.B.K., *Use of the Symbolic Addition Method in the Solution of Centrosymmetric Structures: Description of a Computer Program to Automate this Process*, III, 1969, 100pp.
5. Manacher, G.K. and Dewar, R.B.K., *The UNCLL List-Processing Language: A Preliminary Description*, ICR QUARTERLY REPORT 13, University of Chicago, 1967.
6. Manacher, G.K. and Dewar, R.B.K., *The UNCLL Plex-Processing Language: Preliminary Design Description II, and Tentative Programmer's Manual*, ICR QUARTERLY REPORT 15, University of Chicago, 1967.
7. Fleischer, E.B. and Dewar, R.B.K., *The Solution of the Structure of Tribenzotolarene by Automated Symbolic Addition in the Spacegroup in P212121*, ACA Winter Meeting, Tucson, AZ, Abstract J3, February 1968.
8. Fleischer, E.B. and Dewar, R.B.K., *Structure of a Reduce Nickel (II) Tetradentate Macrocycle*, Nature, 222, 1969.
9. Chock, P.S., Dewar, R.B.K., Halpern, J., and Wong, Lai-Yoong, *The Reaction of Pentacyanocobaltate (II) with Hydrogen Peroxide, Hydroxylamine, Cyanogen Iodine*, JASC 91, 1969.
10. Dewar, R.B.K. and Rosen, P., *CALCTAN (Description and Manual)*, ITT, 1968, 80pp.
11. Dewar, R.B.K., Hochsprung, R. and Worley, W., *The ILTRAN Programming Language*, CACM, October 1969.
12. Fleischer, E.B. and Dewar, R.B.K., *The Structure of Tribenzotolarene*, Acta Cryst., 1971.
13. Dewar, R.B.K., *A Series of Programs for Direct Methods*, paper given at 1969 meeting of international summer school in computing methods in crystallography, Toronto, Canada, published in proceedings, Munkagaard, pp. 63-64, 1970.
14. Dewar, R.B.K. and Belcher, K., *SPITBOL (Description and Programmer's Guide)*, published by Bell Telephone Laboratories, memorandum S4D23, 1971, 90pp.
15. Dewar, R.B.K. and Gray, K.B., *Axiomatic Characterization of the Time-Weighted Rate of Return*, Management Science, Vol. 18, No. 2, October 1971.

16. Dewar, R.B.K. and Lykos, P.G., *The ITT Computation Laboratory*, EDUCOM, Vol. 6, No. 3, Fall 1971.
17. Dewar, R.B.K. and Coulter, C., *Application of Tangent Formula in Protein Crystallography*, Acta Cryst., September, 1971.
18. Dewar, R.B.K., *A Stable Minimum Storage Sorting Algorithm*, Information Processing Letters, Vol. 2, No. 6, April 1974.
19. Dewar, R.B.K. and Belcher, K., *SPITBOL Newsletter's #1, #2, #3, #4, #5*, IIT, published in Dec. 1971, April 1972, June 1972, March 1973, and March 1974, respectively.
20. Dewar, R.B.K. and Buroff, S., *The ITT ALGOL-68 Compiler: Superlanguage Features*, paper given at 1974 International Conference on ALGOL-68 Implementation, University of Manitoba, Winnipeg, Canada, published in proceedings.
21. Dewar, R.B.K. and Buroff, S., *Techniques for Independent Compilation of ALGOL-68 Program Segments*, paper given at 1974 International Conference on ALGOL-68 Implementation, University of Manitoba, Winnipeg, Canada, published in proceedings.
22. Dewar, R.B.K., *Indirect Threaded Code*, CACM, June 1975.
23. McCann, A.P. and Dewar, R.B.K., *1900 SPITBOL Technical Report No. 55*, University of Leeds, December 1974.
24. Dewar, R.B.K., Belcher, K. and Cole, J., *UNIVAC/SPITBOL*, Version 1.0, IIT, March 1975.
25. Dewar, R.B.K., Grand, A., Schonberg, E., and Schwartz, J., *The SETL Data Structures*, SETL Newsletter No. 189, November 1976. Courant Institute, New York University.
26. Dewar, R.B.K. and McCann, A.P., *MACRO SPITBOL - A SNOBOL4 Compiler*, Software Practice and Experience, Vol. 7, 1977.
27. Dewar, R.B.K., Belcher, K. and McCann, A.P., *SPITBOL Implementation*, proceedings of 5th III Conference Guide1, France IRIA, 1977.
28. Dewar, R.B.K., Schwartz, J.T., et al., *SETL as a Tool for Generation of Quality Software*, proceedings of Conference on Software Quality; Novosibirsk, North Holland, 1977.
29. Dewar, R.B.K. and Schwartz, J.T., et al., *Programming by Refinement as Exemplified by the SETL Representation Sublanguage*, TOPLAS, 1, 1, pp 27-49, July 1979.
30. Dewar, R.B.K. and Schonberg, E., *The Elements of SETL Style*, proceedings of ACM 1979 National Conference, Detroit.
31. Dewar, R.B.K., Fisher, G. and Schonberg, E., et al., *The NYU-Ada Translator and Interpreter*, invited paper at IEEE COMPSAC Conference, Chicago, 1980.
32. Dewar, R.B.K., Fisher, G. and Schonberg, E., et al., *Experiences with the NYU-Ada Translator and Interpreter*, SIGPLAN Symposium on the programming language Ada, December 1980.
33. Dewar, R.B.K., Grand, A., Schonberg, E., and Schwartz, J., *Higher Level Programming*, MSS 550pp., Introduction to the Use of the Set Theoretic Language SETL, Courant Institute of Mathematical Sciences, NYU, July 1981.
34. Dewar, R.B.K., Sharir, M., and Weixelbaum, E., *Transformational Derivation of a Garbage Collector*, TOPLAS, October 1982.
35. Dewar, R.B.K., Sharir, M., and Merritt, S., *Some Modified Upsequence Algorithms*, Acta Informatica, May 1983.
36. Dewar, R.B.K., Kruchten, P. and Schonberg, E., *Prerequisites for a Formal Definition of Ada*. Position paper to the Ada-Europe working group on formal definitions, Brussels, December 9, 1983.
37. Dewar, R.B.K., Flynn, S. and Schonberg, E., *Memory Allocation Strategies for Ada on Hierarchical Memory Multiprocessors*. Proceedings of the 1989 Ada-Europe Conference, June 1989.
38. Dewar, Robert with Frank Betz, Fisher, D., Gutlag, J., Hudak, P., Wand, M., Balzer, R. (chairman), Gabriel, R. (editor), Draft report on *Requirements for a Common Prototyping System*, November 1988.

39. Dewar, R.B.K., *Shared Variables and Ada 9x Issues*, Ada 9X Special Report, Software Engineering Institute, January 1990.
40. Dewar, R.B.K., *The Fixed Point Faculty in Ada*, Ada 9X Special Report, Software Engineering Institute, February 1990.
41. Dewar, R.B.K., *Uniformity Issues in Ada 9X*, Technical report from ISO TC2/WG9/URG, preliminary version published in November 1990.
42. Dewar, R.B.K., *Proposal for IEEE bindings for Ada 9X*. Preliminary version submitted to ISO TC2/WG9/NRG (Numerics Rapporteur Group) February 1990.
43. Goodenough, John (ed), Dewar, R.B.K., Evan, Art, Brosgol, Benjamin. *Ada 9X Requirements*, Ada 9X Project office, December 1990.

BOOKS

Computers in the X-Ray Phase Problem
 PhD Thesis, 1968
 Robert B. K. Dewar
 University of Chicago, 1968

IITRAN Programmer's Manual and Language Description
 Robert B. K. Dewar
 Four Pi, Inc., 1971.

CALC'TRAN Description and Programmer's Guide
 Robert B. K. Dewar
 Four Pi, Inc., 1971

COMVEST Users Guide
 Analysis of Pension Fund Investment Performance
 Robert B. K. Dewar, Kenneth Gray, and Larry Fisher
 Banking Administration Institute, 1969

Programming with Sets: An introduction to SETL
 Robert B. K. Dewar, Jack Schwartz, J. Dubinsky, Edmond Schonberg
 Springer-Verlag, 1986

Macro-SPITBOL, the High Performance SNOBOL4 Language
 Robert B. K. Dewar, Mark Emmer, Edward Quillen
 Catspaw, Inc. 1989

Microprocessors: A Programmer's View
 Robert B. K. Dewar, Matthew Smosna
 McGraw Hill, 1990

Impact of Innovative Architectures
 Robert B. K. Dewar
 Chapter in Aerospace Software Engineering
 Edited by Christine Anderson
 American Institute of Aeronautics and Astronautics Inc. 1991

GNAT Users Guide

Robert B. K. Dewar, et al
First version published by Silicon Graphics, 1995
Subsequent editions published by Ada Core Technologies, 1996-2000

GNAT Programmers Reference Manual
Robert B. K. Dewar, et al.
First version published by Silicon Graphics, 1995
Subsequent editions published by Ada Core Technologies, 1996-2000

OTHER PUBLICATIONS

Series of ten articles on various micro-processor architecture issues and UNIX software maintenance issues, Unix Today, 1991-1992 (published with Matthew Smosna, as regular featured columnists on these issues).

Numerous articles on the use and importance of Ada in various trade magazines.

PROFESSIONAL ACTIVITIES (some of the dates are approximate)

ACS (American Chemical Society) National Lecturer, 1969 (lectured on computational issues in modern chemical research).

Member of the Association for Computing Machinery

Permanent member of IFIP WG2.1 (working group on Algorithmic languages)

Chairman of IFIP WG2.1, (1985-1991)

Distinguished Reviewer for Ada 83 development (1980-1983)

Member of the Ada Board (Federal Advisory Panel, advised Ada Joint Program Office on Ada Policy, 1986-1991)

Distinguished Reviewer for Ada 9X development (1989-1995)

Member of ISO (International Standards Organization) WG9 ARG (Ada Rapporteur Group – the Ada language maintenance committee). 1985-present

Chairman of ISO WG9 URG (Ada Uniformity Rapporteur Group), 1990-1993

Chairman of ISO WG9 CRG (Ada Character Issues Rapporteur Group), 1994-1995

Member of ISO WG8 NRG (Ada Numerics Rapporteur Group), 1991-1994

Member of the U.S. Delegation to ISO WG9 (Working Group on Ada), and member of the U.S. TAG (technical advisory group).

ACM/SIGAda National Lecturer, giving talks at SIGAda chapters on Ada (1989-1992)

Track Chairman (education track) for Tri-Ada conference, 1991

Member of Program Committee for IFIP TC2/WG2.1 Working Conference on Constructing Programs from their Specifications (1988)

Member of the DARPA Prototech Executive Committee, advising DARPA on issues in design of prototyping languages, 1990

Project Director for Ada 9X implementation Analysis Team (IAT), 1998-1991

Member of Ada 9X ACVC Review Committee (1993-1995)

Member of the Ada 9X Executive Committee (1992-1995)

Member of the AVO Fast Reaction Team (advises Ada Validation Organization on Ada validation policy issues), 1983-1998

Member of the ACATS Fast Reaction Team (successor to AVO Fast Reaction Team), 1998-present.

Member of the Ada Resource Association (representing Ada Core Technologies), 1998-present.

SOFTWARE EXPERIENCE

(includes only software marketed commercially where Robert Dewar was a primary or the only author)

1. Scientific Applications

- a. MAGIC. Program for automated solutions of centrosymmetric X-ray crystal structures. R.B.K. Dewar and A. Stone (1967). FORTRAN 14,000 lines.
- b. PLR. Data reduction program for the PAILRED X-ray data collection machine. R.B.K. Dewar (1967) IBM-7094 Assembly, 7000 lines.
- c. PPNC. Fourier transform and contour package. R.B.K. Dewar (1967) FORTRAN, 4000 lines.
- d. SORFLS. Crystollographic Least Squares. R.B.K. Dewar and M. Bennett (1968) FORTRAN, 4000 lines.
- e. SORFFE. Crystollographic error analysis. R.B.K. Dewar (1967) FORTRAN, 4000 lines.

Note: These crystallographic programs were widely distributed to industry and universities throughout the world, and some of them are still in use in modified forms today.

2. Commercial Applications

- a. COMVEST. Analysis of Pension Fund Investment Performance. R.B.K. Dewar and L. Fisher (1969), FORTRAN, 8000 lines. Marketed to U.S. banks and trust funds by the Banking Administration Institute, Chicago, IL.
- b. LVDF. Distributed screen-forms oriented data collection system for the INCOTERM SPD 10/20. R.B.K. Dewar (1974), SPD assembly language, 15,000 lines, written for Vickers Ltd. and Incoterm Ltd., UK.
- c. DBMS/DBINQ. Data base management and inquiry program for the Honeywell SPD 20/40 series machines, marketed as part of the RDE (remote data entry) system by Honeywell UK, R.B.K. Dewar (1975), SPD assembly language, 12,000 lines.

- d. SGDEM. Demonstration data entry program for banking industry. Written for Transac, Paris, France as part of the requirements for the Societe Generals banking system. R.B.K. Dewar (1975), SPD assembly language, 6,000 lines.

3. Compilers

- 3a. UNCLL (University of Chicago Low Level List Processing Language). Preliminary compiler for IBM 7094. R.B.K. Dewar (1967), FAP Macro, 5000 lines.
- 3b. CALCTAN (Conversational Desk Calculator Programming Language). Interpreter for IBM 350, R.B.K. Dewar and B. Rosen, IBM Assembly, 4000 lines. Marketed by Four-Pi Systems, Chicago, IL.
- 3c. IITRAN/360. Student language used at Illinois Institute of Technology for teaching beginning programming, and used by over 50 schools and colleges in the Chicago area as part of an NSF sponsored project. R.B.K. Dewar, R. Hochsprung and W. Worley, (1965-1969). IBM 360 assembly language, 20,000 lines.
- 3d. SPITBOL. Fast SNOBOL 4 compiler for IBM 360. In use at over three hundred universities and industrial sites. R.B.K. Dewar and K. Belcher (1969-1973) IBM 360 assembly language, 30,000 lines.
- 3e. IITRAN/1108. Version of IITRAN for UNIVAC 1108. R.B.K. Dewar and R. Schlesinger (1970-1971), 1108 assembly language, 15,000 lines.
- 3f. IITRAN/9400. Version of IITRAN for UNIVAC 9400, written under contract to UNIVAC for use in the Montreal School System, R.B.K. Dewar and R. Schlesinger (1970), 9400 assembly language, 22,000 lines.
- 3g. SPITBOL/1108. Version of SPITBOL for the UNIVAC 1108. R.B.K. Dewar and K. Belcher, 1108 assembly language, 20,000 lines.
- 3h. MACRO SPITBOL. Version of SPITBOL written in a specially designed machine independent macro system. R.B.K. Dewar and A.P. McCann (1974-1977), MINIMAL, 24,000 lines. This system has since been translated for over 20 different processors and is in use at over 1000 sites.
- 3i. SETL. Translator and Interpreter for the Set Theoretic Language invented at NYU by J. Schwartz, R.B.K. Dewar, A. Grand and S. Freudenberger (1976-1982), LITTLE, 100,000 lines.
- 3j. NYU Ada/Ed. This was the first validated Ada translator. Written under contract to the U.S. Army at Fort Monmouth, distributed by the National Technical Information Software Exchange, in use at over 300 sites. R.B.K. Dewar, G. Fisher, E. Schonberg, et al., SETL 20,000 lines.
- 3k. Realia COBOL ANSI 74 COBOL compiler for the Intel 8086, written with K. Belcher and M. Sokol of Realia, Inc. under contract to Transac, Paris, France. Released for IBM PC 3rd quarter of 1983. Passes 3 and 4 (COBOL, 40,000 and 20,000) and the run time library (8086 Assembly language, 80,000 lines) were written by R.B.K. Dewar.

3l. Realia COBOL debugger. One of the first visual debugger technologies. Written for Realia Inc, and used by all their COBOL customers, 8086 assembler, 15,000 lines. 1984.

3m. Alslys Ada compilers for 80x86 and Power PC. Consulted on all phases of design and wrote the back end FINAL phase (backend optimizer, Ada 30,000 lines), and much of the runtime library (8086 assembler, 10,000 lines), 1985-1991.

3n. Ada Verify. Extremely fast Ada syntax checker, distributed with Alslys Ada compilers (Ada 20,000 lines), 1988.

3o. Floating Point Simulator for Alslys Ada compiler. Complete floating point simulator packages for 8087 and 80287 architectures. Used in Alslys compilers on machines with no hardware floating point. 1991, Assembler 4000 lines.

3p. The GNAT compiler, a complete compiler technology for Ada 95 (still the only fully validated Ada 95 compiler). In use at major corporations throughout the world, including Lockheed, Boeing, Raytheon, CSC etc. Robert Dewar was one of the primary authors. The entire system is approximately 400,000 lines of Ada, with Robert Dewar being the primary contributor for at least one third of the total system.

3q. SPITBOL interface for GNAT, a complete interface to SPITBOL pattern matching for use with Ada 95, distributed as part of the GNAT runtime (Ada 95, 4000 lines).

4. Operating Systems and Executives

4a. IMS. Student Monitor System for UNIVAC 1108, R.B.K. Dewar and R. Schlesinger (1970), 1108 Assembly, 2000 lines.

4b. COS 10/20. Assembler and cassette operating system for the INCOTERM SPD 10/20. Written under contract to Incoterm Corp. R.B.K. Dewar (1971-1972) SPD assembly, 12,000 lines.

4c. SPD/DOS. Disk Operating System for Incoterm SPD 10/20 and 20/20. Written under contract to Incoterm Corp., in use at over 1000 sites. R.B.K. Dewar (1971-1972), SPD assembly, 40,000 lines.

4d. PD/FMS. Disk Operating System and File Management System for Incoterm SPD 20/40, written under contract to Incoterm Corp., in use at over 200 sites. R.B.K. Dewar (1975-1977), SPD assembly, 80,000 lines.

4e. EXEC/15, DOS/15. Executive and Disk Operating System for Honeywell SPD 15/25 system, written under contract to Honeywell Corporation, R.B.K. Dewar (1977-1979) 8086 assembly, 75,000 lines.

4f. EXEC/88. Real time executive for the Intel 8086 for use on Honeywell ATM machines and SPD XX. R.B.K. Dewar, 8086 assembly language, 5000 lines.

4g. RTX80. Real time executive for Intel 8088, written under contract to F & S Systems Corp. R.B.K. Dewar, 8088 assembly language, 3000 lines.

4h. WFMS,. Modified version of PD/FMS, written for Honeywell Corp., for use on the Winchester based SPD 20/40. R.B.K. Dewar (1980-1983), SPD assembly, 50,000 lines.

5. Other System Software

5a. SPD RBT/900. A series of five remote batch terminal emulators for the Incoterm SPD 10/20, written under contract to Incoterm Corp. R.B.K. Dewar (1972-1973), SPD assembly, 14,000 lines.

5b. Assembler and linker for the Intel 8088 program development system, written under contract to Transac, Paris, France. R.B.K. Dewar (1978), SPD assembly, 26,000 lines.

5c. Assembler and linker for Intel 8088 development systems for use with Honeywell Level-6 development system, written under contract to Honeywell Corp, R.B.K. Dewar (1978), PASCAL, 14,000 lines.

5d. VAX based development system for 8086 (assembler and linker), written under contract to Scott Systems, Boston, MA. R.B.K. Dewar (1982), C, 14,000 lines.

5e. SPACEMAKER and TERMULATOR, commodity software for IBM PC (PC DOS file compression utility and VT-100 emulator), being marketed by Realia, Inc. R.B.K. Dewar (1982-1983), 8088 assembly language, 8,000 lines.

5f. Screen editor for VERSAL S20 machine, written under contract to VERSAL AB (Stockholm, Sweden), R.B.K. Dewar (1982-1983), UCSD PASCAL, 4,000 lines.

5g. DVED, visual screen editor for the IBM PC, Freeware program, R.B.K. Dewar (1983), 8088 assembly language, 10,000 lines.

5h. DUTIL, Dewar Utilities. Collection of utilities distributed as shareware for IBM PC. - includes advanced version of DVED, see 5g, (1986-1990), 8086 assembler, 20,000 lines, C, 5000 lines.

5i. DAS, DED, DDB. Assembler, Editor and Debugger for teaching elementary assembly language programming (1986, COBOL, 5000 lines, 8086 assembler, 15,000 lines).

5j. Assembler and linker for x86 based system F&S Systems Corp, (C 20,000 lines), 1982

OTHER CONSULTING EXPERIENCE

Technical Consulting

1. Served as member of site visit team for computer assisted instruction initiative, NSF 1975
2. Consulted for Mitre Corp, Washington, D.C on the design of the operating system to support the TICCET Computer Assisted Instruction system, developed with NSF support, 1975.

3. Consulted for Incoterm Corporation on design and development of their intelligent terminal systems, 1972-1977.
4. Consulted for Honeywell Corporation on design and development of intelligent terminal systems, 1977-1981
5. Consulted for MacDonald Automation, Hospital Services Division, St. Louis, MO on design of COBOL application programs for file handling, including interactive data entry and batch processing functions, 1980-1981.
6. Consulted for Realia Inc on general COBOL design issues, also worked with many Realia customers, helping them to use Realia COBOL to design and implement their applications, 1982-1985
7. Consulted for F&S systems, on design of operating system and utilities for their x86 based products, 1982.
8. Consulted for Intermetrics Corp, on the design of the Red Language (one of the competing designs for the Ada project), 1981.
9. Consulted for Transac SA, Paris, France, on the design of their new operating system for the SPD 27 system (the first commercial system in the world based on the 8086 architecture from Intel), with particular emphasis on file handling and transaction handling (1980-1982).
10. Consulted for VERSAL AB, Stockholm, Sweden, on the design of S10, S20, and S30 machines. This involved the design of a new microprocessor architecture specialized for execution of UCSD Pascal (1982-1983).
11. Consulted for Alsys Inc, on the design of Ada compiler technology, 1986-1993.

Professional Recognition

1. Acknowledgment from President of ACM (President's Letter, Communications of the ACM, December 1987). "examples abound of significant programs, programs of enormous practical importance, that were written by individuals: Thompson and Ritchie's original UNIX, Bricklin and Fylstra's Visicalc, Knuth's TEX, Dewar's Realia COBOL, and Reid's Scribe"
2. OUTSTANDING Ada COMMUNITY CONTRIBUTIONS, Awarded by ACM SIGAda to Robert Dewar, 1995.
3. Certificate of Appreciation for contributions to Ada 83 development, signed by the Undersecretary of Defense (approximately 1993)
4. Commemorative Award for contributions to the completion of the Ada 95 standard, from the Ada Joint Program Office of the department of defense, 1996.

Legal Consulting

Cases where Robert B. K. Dewar was retained as expert witness and testified)

1. National Data Communications vs. St. Mary's Hospital. Case involving software copyright issues, trial in US Federal Court, Texas 1982.
2. 580 F.Supp.474. Selden vs. Honeywell, 1984. Case involving allegations of fraud in connection with delivery of operating system software. Trial in New York Federal Court.
3. 1998 WL 800344, Intergraph vs. Bentley, 1998. Case involving software copyright and licensing issues. Trial in Philadelphia Federal Court.
4. GEAC vs. GRACE, 1997-2000. Case involving software copyright issues. Trial in Newark Federal Court.
5. Akamai vs Cable and Wireless, 2003-2004, Case involving patents for internet content distribution methods. Trial in Boston Federal Court.

Other Recent Cases where Robert B. K. Dewar was retained as expert witness

6. Advanced Micro Devices, Inc. and AMD International Sales & Service, Ltd. vs Intel Corporation and Intel Kanushiki Kaisha, 2006-2007. Antitrust action. My involvement was in the issue of Intel adjusting their compiler to be disadvantageous to AMD. Reports submitted, case settled before trial.
7. Altnet, Inc., et al v RIAA, Overpeer, and Mediasentry, 2006-2008. Patent case brought against RIAA for alleged infringement of patents relating to hashing. Reports submitted, case settled before trial.
8. Cambridge University Press, Oxford University Press, Inc., and Sage Publications, Inc vs Mark P. Becker, in his official capacity as Georgia State University President, et. al. Suit involving alleged copyright infringement by university of Georgia in their management of their online reserve system. Report submitted, deposed, but did not testify.

Robert Dewar's Contributed Articles
March 2005 – November 2010

(Sorted by publication date; oldest first. Items in red indicate European publications)

Published Articles

| Publication | Topic/Title | Author | Status | Web Link |
|----------------------------------|---|-----------------------------------|---|--|
| <i>Military Embedded Systems</i> | "FLOSS helps you take control of your bytes" | Robert Dewar | Published in September 2005 e-Letter. Also published in VMEbus Systems October 2005 Issue and MES Spring 2006 Issue. | http://www.mil-embedded.com/articles/id/?820 http://www.vmecritical.com/articles/id/?820 |
| <i>COTS Journal</i> | "Ada 2005 Strengthens Ada's Safety-Critical Muscles" | Robert Dewar | Published in November 2005 Issue | http://www.cotsjournalonline.com/home/article.php?id=100424 |
| <i>VMEBus Systems</i> | "Using static analysis tools for safety certification" | Robert Dewar and Ben Brosgol | Article published in April 2006 issue. | http://www.vmecritical.com/articles/id/?2 |
| <i>Embedded Systems Europe</i> | "Safety-Critical Design for Secure Systems" | Robert Dewar | European publication. Article published in June 2006 issue. | < http://i.cmpnet.com/embedded/europe/esejun06/esejun06p35.pdf > |
| <i>ECN Magazine</i> | "Ada for Embedded Programming" | Robert Dewar and Jose Ruiz | "Web-exclusive" article published online in September 2006 and was also cross-promoted in September print issue. | http://www.ecnmag.com/article.aspx?id=122782&terms=AdaCore |
| <i>Cross Talk</i> | "Integrated Quality Assurance for Evolutionary Multi-Platform Software Development" | Robert Dewar (Ben Brosgol) | Ben revised September 2002 article for online publication. "Web-exclusive" article was published on October 30, 2006. | http://www.stsc.hill.af.mil/crosstalk/2006/11/0611dewar.html |
| <i>Military Embedded Systems</i> | "Building secure software: your language matters!" | Robert Dewar/Rod Chapman (Praxis) | Published online and in the Winter 2006 issue. | http://www.mil-embedded.com/PDFs/AdaCore.Win06.pdf |
| <i>SD Times</i> | Guest View - "Why high-security systems require open-source software" | Robert Dewar | Published online on January 15, 2007. | http://www.sdtimes.com/article/column-20070115-01.html |
| <i>EE Times</i> | Letter to the Editor - "Ada vs. Java vs. C/C++" | Robert Dewar | Published online and in July 2, 2007 issue. | http://www.eetimes.com/showArticle.jhtml;jsessionid=5VPP |

| Publication | Topic/Title | Author | Status | Web Link |
|---|--|-------------------------------------|---|---|
| | | | | 25S4VFDM2QSNLPSKHSC JUNN2JVN?articleID=200001 095 |
| <i>Electronic Design</i> | Point of View – "Multithreading: It's not new!" | Robert Dewar | Published online and in the April 18, 2007 ED Update e- mail newsletter. | http://www.elecdesign.com/Ar ticles/Index.cfm?ArticleID=15 410 |
| <i>Doctor Dobb's Journal</i> | "Alia Vox" Guest Editorial - "Safety AND Security" | Robert Dewar | Published online May 7 and in the June 2007 print issue. Also published online on Embedded.com May 7, 2007. | http://www.ddj.com/dept/secu rity/199300140 http://www.embedded.com/sh owArticle.jhtml?articleID=199 502018 |
| <i>COTS Journal</i> | "Ada "Reloaded" a Winner for High-Integrity Real- Time Apps" | Ben Brosgol and Robert Dewar | Published online and in July 2007 print issue. | http://www.cotsjournalonline.c om/home/article.php?id=1006 87 |
| <i>SD Times</i> | "Letters to the Editor - Ada's Tried and Tested" | Robert Dewar | Robert's response to Jennifer deJong's Safety-critical Java Article was published October 12, 2007. | http://www.sdtimes.com/ar ticle/opinion-20071001- 03.html |
| <i>CrossTalk</i> | "Computer Science Education: Where Are the Software Engineers of Tomorrow?" | Robert Dewar and Ed Schonberg | Published online in January 2008 issue. Article generated significant buzz in online news sites and blogs worldwide. | http://www.stsc.hill.af.mil/Cros sTalk/2008/01/0801DewarSc honberg.pdf |
| <i>Electronic Design</i> | Point of View: "Use Ada for Better Safety, Security, and Reliability" | Robert Dewar and Ben Brosgol | Managing Editor Rich Gawel posted the opinion piece online on February 2, 2008. It also ran as part of the ED Update e-mail newsletter and in the Feb. 28 Web Table of Contents in print. | http://electronicdesign.com /Articles/ArticleID/18141/1 8141.html |
| <i>EE Times</i> | Design Feature: "There's nothing new about multi- core mania" | Robert Dewar | Article published online on February 26, 2008 | http://www.eetimes.com/news /design/showArticle.jhtml?arti cleID=206900265 |
| <i>Military Embedded Systems</i> | Legacy Software Migration Column: "Ada Matters!" | Robert Dewar | Article published online on and in May print issue. Debut of MES' new Legacy Software Migration column. | http://www.mil- embedded.com/pdfs/LegacyS WMigration.May08.pdf |
| <i>Embedded Systems Engineering</i> | "Security in High Reliability Applications: Is it safe?" | Robert Dewar | Article was published in the April 2008 print issue and online on May 1, 2008. <i>Paris office re-purposed Robert's SD Times Guest View [F928-013].</i> | http://www.esemagazine.com /index.php?option=com_cont ent&task=view&id=509&Itemi d=2 |
| <i>Embedded Control Europe</i> | "Embedded Insecurity: just how much is at risk?" | Robert Dewar | Article appeared online and in the June 2008 print issue, (page 15). | http://www.embedded- control- europe.com/c_ece_knowhow/ 237/ecejun08.pdf |
| <i>Software Tech News</i> | "Software and Intellectual Property Rights" | Robert Dewar | Article appeared online and in print as part of August 2008 newsletter. | https://www.softwaretechnew s.com/stn_view.php?stn_id=4 5&article_id=110 |

| Publication | Topic/Title | Author | Status | Web Link |
|--|--|---|--|---|
| <i>Embedded.com</i> | "Open Source Software for High Reliability Applications: Is it safe?" | Robert Dewar | Article published online on October 29, 2008. | http://www.embedded.com/design/opensource/211600582?requestid=307145 |
| <i>ECN</i> | "Take a New Look at Ada" | Robert Dewar | Article was published on Ecnmag.com on November 17, 2008. | http://www.ecnmag.com/Embedded-Systems-Take-a-New-Look-at-Ada.aspx?menuid=548 |
| <i>Military Embedded Systems</i> | "Legacy is not a Four Letter Word" | Robert Dewar | Article was published digitally on the MES.com on January 1, 2009. | http://www.mil-embedded.com/articles/id/73729 |
| <i>Military Embedded Systems</i> | "Legacy Software, It's the 'new' thing." | Robert Dewar | Article, was published in the March/April 2009 print issue. | Print only |
| <i>Embedded Computing Design</i> | "Making Static Analysis a Part of Code Review" | Robert Dewar and Tucker Taft (SofCheck) | Article appeared online and in the June 2009 e-Newsletter. | http://www.embedded-computing.com/articles/id/74014 |
| <i>Communications of the ACM</i> | "CS Education in the U.S.: Heading in the Wrong Direction?" | Robert Dewar and Owen Astrachan | Robert's "point" for the point-counterpoint feature was published in the July 2009 print issue and online. | http://cacm.acm.org/magazines/2009/7/32090-cs-education-in-the-us-heading-in-the-wrong-direction/fulltext |
| <i>New Electronics</i> | "FLOSS and COTS: Natural partners for decreasing costs and increasing quality" | Robert Dewar | Article appeared online on August 11, 2009. | http://www.newelectronics.co.uk/article/19442/FLOSS-and-COTS-Natural-partners-for-decreasing-costs-and-increasing-quality.aspx |
| <i>Managing Software Professionals</i> | "Computer Science Education: Where Are the Software Engineers of Tomorrow?" | Robert Dewar and Ed Schonberg | Article appears in a book edited by G. P. Sudhakar. (Sept 2009? – Nancy to check publication date) <i>Article originally appeared in the January 2008 Issue of CrossTalk magazine.</i> | |
| <i>Military Embedded Systems</i> | "Software warranties; A new era?" | Robert Dewar | Article was published in the September 2009 issue | http://www.mil-embedded.com/articles/id/?4217 |
| <i>Elsevier Reference Handbook</i> | Case Study 6.1. "The GNAT Pro Ada Compiler—A Case Study in Achieving High Quality in the Presence of Rapid Change" | Robert Dewar | Case Study appeared in the November 2009 Handbook on Developing Mission-Assured and Safety-Critical Systems. On January 22, 2010 Melanie Benson sent a PDF of the published piece and indicated that Robert will be receiving a hard copy of the book shortly. | |
| <i>Defense Tech Briefs</i> | "Integrating Formal Methods into Software Toolsets for Avionics Certification" | Robert Dewar | Article appeared online and in the August 2010 print issue of Defense Tech Briefs | http://www.defensetechbriefs.com/component/content/article/8238 |

Articles Awaiting Publication

| Publication | Topic | Author | Status |
|--|--|--------------|--|
| <i>Military Embedded Systems</i> | November/December "COTS Software in Critical Systems The Case for Freely Licensed Open Source Software" | Robert Dewar | Rainier submitted Robert's article on July 27 and a revised version with graphics and captions on August 9. Robert's article was originally written for the September Executive Point/Counterpoint issue, but has been moved to the November/December issue and will be featured as a Guest Opinion piece. Currently waiting on publication. |