Software

The actual software may be given patent, copyright, or trade secret protection. A patent or trade secret protects the underlying process or routine implemented via the software—for example, the process that performs the basic calculations and steps that generate the virtual reality world series.

A copyright protects the literal copying of the software. It also protects the software's overall structure or organization, sometimes referred to (with some controversy) as its "look and feel." Simply stated, a copyright protects copying. Therefore, a similar software program independently developed would be outside the scope of the copyright protection afforded to the multimedia developer.

Patent or trade secret protection protects the underlying computer-implemented process much the same way as with computer hardware.

Audio or visual entertainment

A multimedia application may also include audio or visual entertainment presented to the user. For example, in "Virtually Anyone Can Play Ball," the national anthem sung by Whitney Houston at the beginning of the baseball game is subject to copyright protection. Once again, this would prevent another from freely copying the audio or visual entertainment.

Multimedia developers might need permission or a release from performers to use the audio or visual entertainment. For example, the multimedia developer in the virtual reality world series would need permission from Whitney Houston to play her version of the national anthem.

Interactive user-interface mechanism

A multimedia application's interactive user-interface mechanism generally comprises software to drive the interface, audio or visual interfaces to communicate with the user, and hardware to provide a user-friendly, realistic experience.

In addition, however, the user interface provides an opportunity for exploiting trademark protection. In particular, the user interface represents the direct connection or communication with the user. This interface can therefore promote the "good will" of the multimedia application and manufacturer. It can be used to present catchy words or phrases to reveal the source of the multimedia application. This helps users identify with the quality of the multimedia application for future releases or versions, as well as assist in the sale of other multimedia-related products. Thus, trademarks are important intellectual property assets.

"So, you see," Rex concluded, "your new multimedia application can be given trade secret, patent, copyright, and trademark protection. Each type of intellectual property protection has advantages and disadvantages. But it's up to you to assess what makes sense in accordance with your business objectives and budgetary constraints."

"Let me think about my options and the associated costs involved, and I'll get back to you soon," Susan said.

"That's fine," Rex said, "but don't wait too long, or you might jeopardize your intellectual property rights—leaving you with virtually nothing!"

Update

Send news about research, public policy, or professional issues to Bob Carlson, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264; Internet b.carlson@computer.org

Task force to survey software engineers in 1996

Patricia Douglas, *IBM*

One of the factors associated with recognized formal professions, as Capers Jones outlined in "Legal status of software engineering" (*Computer*, May 1995, pp. 98-99), is a well-defined body of knowledge. Often this knowledge includes many subsets of more specialized knowledge. Also necessary is an appropriate academic curriculum to transfer knowledge to students.

In keeping with the above principles, the Joint Task Force on Software Engineering Ethics and Professional Practices, established by the IEEE Computer Society and the ACM, is working to have software engineering recognized as the 37th engineering profession.

The joint steering committee has chartered four task forces. The first task force was established to adopt standard definitions, and this has been completed. The second task force was established to define the required body of knowledge and recommended practices. (In electrical engineering, for example, electromagnetic theory is part of the body of knowledge, while the National Electrical Safety Code is a recommended practice.) The third task force was established to define ethical standards. The fourth task force was established to define educational curricula. (The steering committee has also begun to study certification or licensing in other engineering professions.)

The second task force, for establishing the body of knowledge and recommended practices, is chaired by the author, joined recently by cochair Anthony Cocchi. A senior designer/programmer with IBM Research, Cocchi brings a background in systems architecture, software design, and programming to this work.

Our task force is developing a series of surveys to incrementally define the body of knowledge and practices that apply to different skill levels needed at different points in the career of a software engineer. These surveys will identify knowledge components falling into two categories: generic knowledge (such as mathematics and science engineering) and specific knowledge (for example, software analysis and software architecture). In each area, we will address static as well as dynamic knowledge.

We began focus groups in early 1994 to define a list of possible areas for consideration. The results were organized for inclusion in an international survey. The survey methodology was chosen because it enabled the most farreaching industry coverage. In addition, the survey format will let us analyze areas of commonality in worldwide software engineering as well as unique aspects of a given industry segment.

The survey will consist of statements concerning the tasks performed by professionals at different levels of expertise (see sidebar called "Sample survey questionnaire"). Our goal is to complete this phase by the end of October

We are currently identifying industry subchairs who will have the following responsibilities:

- Serve as a project manager within an industry sector to see that task items are completed on schedule and returned to the chair. Sectors include transportation, consumer products, engineering/design, medical/health, education, insurance, pharmaceutical, telecommunications, aerospace/defense, public utility, financial/banking, state and local government, manufacturing, library, legal, publishing, general business, federal government, and other.
- Arrange for the task writing workshop with respect to location, mailing notices, and hosting.
- Identify and work with key individuals to assist in distributing the survey and collecting the completed survey.
- Serve as a communications point for work in progress.
- Identify a mechanism within an industry sector to publicize ongoing work and gain support.

We expect to have the survey created by the end of October and conducted early in 1996. So far, we have identified only 58 topics for survey questions, which is about 10 percent of what we will eventually require. If you can participate as a subchair in this work, please contact Patricia Douglas at pjd@vnet.ibm.com; or Tony Cocchi at tony@watson.ibm.com.

Sample Survey Questionnaire

Table A is an abbreviated survey questionnaire, formatted as it might be for the actual survey instrument to illustrate the kinds of questions that we need to create. When we compile and evaluate the data after the survey has been distributed, we plan to illustrate the results both in tabular and graphic form for publishing purposes.

Table A. This table reports the percentage of respondents to our beta survey (*n* = 68) who indicated that "a software professional in my organization should be able to" do each of the seven tasks listed, for each of four job categories.

	Object-Oriented Programming Percentage responding "yes" to each of seven questions for five categories	Novice software engineer (1-3 years)	Experienced software engineer (7-10 years)	Software project manager	Specialist (scientist, domain engineer)	Not applicable in my organization
1.	Define "object," "class," inheritance," and "polymorphism."	79.4	86.8	55.9	70.6	2.9
2.	Read and understand programs written in an object-oriented language.	73.5	86.8	32.4	57.4	5.9
3.	Write simple object-oriented programs.	80.9	85.3	11.8	55.9	5.9
4.	Maintain object-oriented programs.	58.8	83.8	8.8	41.2	5.9
5.	Select appropriate classes from a class library and define appropriate subclasses for a given application.	27.9	83.8	8.8	57.4	7.4
6.	Create an application framework and class library for a particular problem domain.	8.8	70.6	10.3	67.6	5.9
7.	Evaluate object-oriented programs and identify opportunities for improvements.	5.9	76.5	13.2	57.4	7.4

In the object-oriented programming area, it is clear that the novice software engineer is generally expected to be able to perform the tasks representing the lower levels of Bloom's taxonomy, but this decreases noticeably at higher taxonomic levels. This is true (although less markedly so) for experienced software engineers. For the software project manager, little is generally required in object-oriented programming. Certain interesting conclusions can be drawn from the data. For example, the specialist shows a dip in percentage for statement # 4 ("Maintain object-oriented programs"). Evidently program maintenance is more likely to be left to experienced software engineers.