School of Law
UNIVERSITY OF GEORGIA
Prepare.
Connect.
Lead.

Journal of Intellectual Property Law

Volume 9 | Issue 1 Article 3

October 2001

XML Schemas and Computer Language Copyright: Filling in the Blanks in Blank Esperanto

Douglas E. Phillips
Washington, D.C.

Recommended Citation

Douglas E. Phillips, *XML Schemas and Computer Language Copyright: Filling in the Blanks in Blank Esperanto*, 9 J. INTELL. PROP. L. 63 (2001).
Available at: https://digitalcommons.law.uga.edu/jipl/vol9/iss1/3

XML SCHEMAS AND COMPUTER LANGUAGE COPYRIGHT: FILLING IN THE BLANKS IN BLANK ESPERANTO

*Douglas E. Phillips**

TABLE OF CONTENTS

* The author is a lawyer in Washington, D.C. The views in this article are solely his own. The author gratefully acknowledges the helpful comments of David Barg, Edmond Phillips, J. Gregory Swift, and Carolyn Taylor.

I. INTRODUCTION

> *"Suppose some one devised a set of words or symbols to form a new abstract speech, with inflections, but as yet with no meaning, a kind of blank Esperanto."*
>
> Learned Hand, 1921[1]

XML, the Extensible Markup Language, "has taken the computer world by storm."[2] One of the countless XML primers on computer bookshelves proclaims: "XML promises to transform the basic structure of the Web, moving beyond HTML and replacing it with a stronger, more extensible architecture."[3] Bill Gates, chairman of Microsoft Corporation, calls XML a "breakthrough technology."[4] According to Tim Berners-Lee, inventor of the World Wide Web, XML is one of two main technologies that will create the "Semantic Web," which "will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users."[5]

XML is not itself a complete markup language. Rather, XML has been called a language for *creating other languages*.[6] More specifically, XML is a syntax that allows computer users to create their own sets of markup tags, also known as "schemas."[7] The "most widely used markup language in the universe" is HTML, the HyperText Markup Language.[8] HTML is used primarily to control the appearance of documents on the World Wide Web.[9] XML, in contrast, enables users to specify markup tags that identify

[1] Reiss v. Nat'l Quotation Bureau, Inc., 276 F. 717, 718 (S.D.N.Y. 1921).

[2] BRIAN E. TRAVIS, XML AND SOAP PROGRAMMING FOR BIZTALK SERVERS xiii (2000). SOAP, the Simple Object Access Protocol, "is an Internet Engineering Task Force (IETF) Internet draft that lets you invoke procedures on remote systems by using a standard set of XML tags." *Id.* at 18.

[3] SIMON ST. LAURENT, XML: A PRIMER 1 (2d ed. 1999).

[4] TRAVIS, *supra* note 2, at xiii.

[5] Tim Berners-Lee et al., *The Semantic Web: A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities*, SCI. AM. 34, 36 (May 2001). Like any computer trend, XML also has its skeptics. *See* Peter Lucas, *Pumped Up: Business Reaches for the Promise of XML*, ELECTRONIC COM. WORLD, April 2001, at 25 ("When Extensible Markup Language debuted as a data exchange language in the late 1990s, it did not take long for chief technology officers to discover that the initial hype surrounding XML was overblown.").

[6] ELIZABETH CASTRO, XML FOR THE WORLD WIDE WEB 35 (2001).

[7] TRAVIS, *supra* note 2, at 1, 40-42.

[8] *Id.* at 35. Travis cites estimates "that 100 billion HTML pages are on the Web." *Id.*

[9] ST. LAURENT, *supra* note 3, at 10 ("HTML was doomed to life as a formatting language instead of a structured framework for documents").

document elements according to their content.[10] Then, with style sheets, XML can be used to serve documents on the Web, or otherwise exchange documents over networks, so that the documents not only appear as intended but also allow search engines and software agents to locate and process particular types of content by referring to the specified tags.[11]

XML, in other words, is something like the "blank Esperanto" that Learned Hand imagined—with XML's syntax specifying the "inflections," and XML schemas filling in the blanks. Hand concluded that the core set of words, symbols, and inflections, even if meaningless, could be protected by copyright.[12] But how does such a view hold up today, eighty years later? And how does it apply to copyright in the schemas that extend XML's vocabulary and give meaning to XML documents?

The XML specification is an open standard set by the World Wide Web Consortium (W3C), a group of companies that includes software industry leaders.[13] The copyright status of XML schemas, however, is an open question. One XML entrepreneur states: "XML schemas are becoming the intellectual property of e-business . . ."[14] Although some authors of XML schemas have explicitly dedicated them to the public domain,[15] others treat

[10] *Id.* at 15. HTML can be written as XML using the XHTML specification. TIM BERNERS-LEE & MARK FISCHETTI, WEAVING THE WEB: THE ORIGINAL DESIGN AND ULTIMATE DESTINY OF THE WORLD WIDE WEB BY ITS INVENTOR 162 (1999).

[11] *See* ST. LAURENT, *supra* note 3, at 1 (introducing the capabilities of XML).

[12] *Reiss*, 276 F. at 718, 719.

[13] *See* W3C Intellectual Property FAQ, *at* http://www.w3.org/Consortium/Legal/IPR-FAQ-20000620.html (last visited Aug. 30, 2001) (explaining the facets of the standard). Tim Berners-Lee, now the Director of W3C, has described his relief when CERN, the European Particle Physics Laboratory at which he invented the Web, agreed in 1993 to place the Web protocol and code in the public domain. BERNERS-LEE, *supra* note 10, at 74.

[14] *Extensibility's XMLschema.com Goes Live; New Site Provides Ubiquitous Support for Managing and Exchanging XML-Based E-Business Grammars*, BUS. WIRE, Dec. 8, 1999, LEXIS, Nexis Library, News Group File, All. XML schemas may also embody business rules or processes that companies consider their intellectual property. According to a news story, "Extensibility's XML Authority, a software package priced at $99.95, allows companies to use existing intellectual property as XML schemas." Christina Dyrness, *Local Firm a Leader in Web Field*, RALEIGH NEWS AND OBSERVER, May 19, 1999, at D1.

[15] *See, e.g.*, Tom Sawyer, *Data Standards Sought*, ENGINEERING NEWS-RECORD, Feb. 12, 2001, at 21 (stating that a plant design and automating consultant said his company is "plac[ing] its own XML schema for control valves and centrifugal compressors in the public domain in a bid to entice others to do the same"); Clive Davidson, *Reuters Increases Commitment to XML*, SEC. INDUSTRY NEWS, Jan. 29, 2001, LEXIS, Nexis Library, News Group File, All (Reuters has contributed NewsML, its XML specification for news stories, to the public domain). Explicitly contributing an XML schema to the public domain at least implies, of course, that the schema would or at least might otherwise be copyrightable.

their XML schemas as proprietary.[16] Still others take a mixed approach. For example, XrML (Extensible Rights Markup Language) is licensed to the public on a royalty-free basis.[17] The license, however, includes various covenants by the licensee, such as an agreement to use reasonable efforts to ensure compatibility of the licensee's products with XrML.[18] Clicking an "accept" button for the license terms is a prerequisite for even viewing the XrML specification.[19]

If a particular XML schema succeeds in fulfilling XML's advertised potential for providing the "*lingua franca* of e-commerce transactions"[20] in a particular industry, or for a particular set of applications, then copyright ownership of the schema can carry with it significant practical influence and economic power. Developing proprietary "extensions" of public domain technology in hopes of gaining effective control of the technology is not an unknown strategy in the technology industry. Extensible Markup Language, by definition, is extensible, and XML schemas are the extensions. Although a copyright owner may be likely to be constrained to grant free or inexpensive licenses to at least some users to make the schema effective, the owner can selectively withhold licenses from, for example, those who want to modify the language, and in other cases the owner can collect substantial royalties for granting license rights.[21]

Because XML schemas are commonly called markup *languages*,[22] the novel question whether XML schemas are potentially protected by copyright—*i.e.*, "copyrightable"—implicates a more familiar but, surprisingly,

[16] "Some vendors are writing proprietary schema, raising data exchange concerns." Sawyer, *supra* note 15, at 21. For example, one news article reports that a company known as printCafe "has developed a new data-exchange protocol . . . based on XML," but the new protocol "is not public domain." Burt Langford, *E-production Is E-volving*, BUSINESS AND MANAGEMENT PRACTICES, FOLIO SPECIAL SOURCE BOOK ISSUE (2001), LEXIS Nexis Library, Company News File, All. Rather, "[i]n one form or another, you have to do business with printCafe to use it . . . ." *Id.*

[17] *See* XrML License Agreement, *at* http://www.xrml.org/xrml_license.stm (last visited Aug. 30, 2001).

[18] *Id.*

[19] *See id.*

[20] TRAVIS, *supra* note 2, at 33.

[21] The interplay of interests will not always be obvious. Some XML schema authors may not need to rely on proprietary rights in a schema—for example, if they supply software (apart from the schema) that is clearly copyrightable and essential in making productive use of the schema—and some suppliers of such software may prefer broad access to schemas authored by others. The analysis in this Article does not attempt to determine which interests might or might not benefit from XML schema copyrightability, but instead focuses on legal issues that warrant attention by all concerned.

[22] *See* CASTRO, *supra* note 6; TRAVIS, *supra* note 2.

still-unresolved question of copyright law. That question is whether computer languages in general are copyrightable. At the same time, XML schemas have particular characteristics that differentiate them from other computer languages and that warrant specific consideration.

This Article, after further describing XML and XML schemas, reviews the general question of computer language copyright. The Article then addresses the particular characteristics of XML schemas that relate to copyrightability. Next, the Article examines the possibility that XML schemas represent a "merger" of expression and idea. The merger doctrine, although deeply embedded in copyright doctrine and history, is not free from confusion. Its application involves a complex process of constructive abstraction. Nevertheless, this doctrine plays a central role in preserving expressive and "receptive" freedom, and the various copyright compromises of recent legislation have not substantially changed it.

The Article concludes that, although software for developing, validating, and using XML schemas ordinarily will be copyrightable, the established copyright principles reflected in the merger doctrine raise significant questions about the copyrightability of XML schemas themselves. The Article also suggests that consideration of the copyrightability of XML schemas may help focus the broader issues presented by the broader question whether copyright protects any computer language.

II. HTML, XML, AND XML SCHEMAS

One reason for the success of the World Wide Web is the relative simplicity of HTML, the language in which Web pages are formatted.[23] To see how XML differs from HTML, and to examine the copyright issues posed by XML-based markup languages, consider an example. Here is a fragment of HTML code defining the body of a hypothetical Web page:

```
<body>
<h1>Welcome to the Dry-as-Dust Martini Page</h1>
<p>Here's all you need:</p>
<ul>
<li>5 ounces gin</li>
```

[23] *See* ST. LAURENT, *supra* note 3, at 10. HTML is an application of SGML, the Standard Generalized Markup Language. *Id.* at 7. XML can be viewed as a simplified version of SGML. *Id.* at 12.

```
<li>1/2 ounce vermouth</li>
<li>1 gin-soaked olive</li>
</ul>
<p>Here's all you do :</p>
<p>Shake.</p>
<p>Drink.</p>
<p>Repeat.</p>
</body>[24]
```

The available elements and attributes in HTML, and the rules that govern potential values of those attributes, are determined by the HTML language specification and, as a practical matter, by browser vendors. As HTML has evolved, browser vendors have added new markup tags. Nevertheless, HTML remains essentially a formatting language, as illustrated by the tags above, and one to which individuals and companies (other than browser vendors) are not generally able to add new tags. The *p* (paragraph) element may contain text on any subject; the *p* tag relates only to how the text will look. The *li* (list item) element may contain items that describe ingredients for a martini, the names of baseball players, or the number of cabins in Montana. HTML does not know, and it does not care.

In XML, the available elements and attributes, and rules governing the values of those attributes, can be defined by anyone. This characteristic is what makes XML "extensible"—anyone can "extend" XML—and it is also what makes XML, by itself, something less than the "markup language" its name denotes. The ultimate markup language is not XML alone but rather the combination of XML and a "document type definition" (DTD) or other "schema" in which someone defines a set of available elements and attributes, using syntax consistent with the W3C XML specification.[25]

[24] An HTML document contains two basic "elements," the *head* and the *body*. Nested within the *body* element are various additional elements that present the content on the browser screen. For example, the *img* element tells the browser to insert an image. The *img* element has an "attribute" called *src*, for source. The "value" of this attribute is the name of the source file for the image. The *h1* element specifies a top-level header, the precise appearance of which is selectable by the user as a browser preference. The *p* element specifies a paragraph. The *ul* element refers to an unnumbered list, and the *li* element refers to a list item. *See generally* W3C, *HTML 4.01 Specification* (Dec. 24, 1999), *at* http://www.w3.org/TR/html4/ (last visited Aug. 30, 2001).

[25] *See* CASTRO, *supra* note 6, at 13.

For example, the HTML document outlined above could be converted to an XML document, but first it would be necessary to create a new set of markup tags (or to define the existing HTML tags in an XML schema). We could call the new markup language defined by our new set of tags HHML, for Happy Hour Markup Language. To take advantage of XML, one would want to create HHML tags that do not merely specify formatting but that have substantive meaning. For example, the *ul* element, rather than being just any unnumbered list, could become the *recipe* element. We could add an attribute for the *recipe* element called *drink*, the value of which would be the drink's name. The *li* elements within the *ul* element could be retagged as *ingredient* elements. In XML, the relevant portion of the code would look like this:

```
<recipe category="martini" drink="Dry-as-Dust Martini">
<ingredient>5 ounces gin</ingredient>
<ingredient>1/2 ounce vermouth</ingredient>
<ingredient>1 gin-soaked olive</ingredient>
</recipe>
```

To be able to use these new elements and attributes, we would need to define them in a DTD or other schema. We could define the element *recipe* and the attributes *category* and *drink* with the following fragment of a DTD:

```
<!ELEMENT recipe (ingredient+)>
<!ATTLIST
   recipe category (martini | margarita | other) #REQUIRED
   recipe drink CDATA #REQUIRED>
```

This DTD fragment defines the element *recipe* as containing one or more instances of the element *ingredient* and as having two attributes, *category* and *drink*, each of which is required to be given a value in any code written with HHML. The *category* attribute must have one of the three values shown in the definition; the *drink* attribute can contain any character data (letters, numbers, etc.). Although the fragment shown here is just that, full-fledged XML schemas are often colloquially called markup languages.

HTML tags, as mentioned earlier, tell a Web browser how to format the contents of a document. XML tags, based on an XML schema, can richly describe the content but not the format. Web designers use separate code known as style sheets, or other formatting technologies, to control the appearance of an XML document.[26] For documents tagged in Happy Hour Markup Language, we could use a style sheet to specify formatting for the *recipe* and *ingredient* elements that would cause them to appear the way we wanted, perhaps exactly as the *ui* and *li* elements appear in HTML. The difference is that, in HTML, we would not have created objects within our Web page that other Web sites or applications could access based on their content. For example, if someone wanted to collect mixed-drink recipes that appear on the Web in HTML pages, it would be necessary to rely on an existing classification by a search engine such as Yahoo! or to look for appearances of various words describing mixed drinks along with the word "recipe." In either case, collecting the recipes would require manually trying to identify, extract, and organize relevant information.

With Happy Hour Markup Language, however, a search engine or software agent could locate Web pages that use HHML and, within those pages, go straight to the drink recipes. Because the *recipe* element includes *category* and *drink* attributes, the values of which are the drink's category and name, it would be easy to assemble a list of recipes by drink categories and names. Other attributes such as degree of dryness, alcohol content, and the like could be added to HHML and additional preset drink category names could be specified when desired. Some attributes could be required and others could be optional. The ingredient elements could be structured to require numeric data for quantities and, separately, ingredient names chosen from a list. Applications or Web services could then be developed that would allow users to choose among various drinks described on various sites and, based on the number of guests expected, to compute shopping lists for relevant ingredients.

Eventually, if enough Web sites relating to mixed drinks elected to adopt Happy Hour Markup Language, or if an industry group could be persuaded to adopt it, HHML could become the "standard" (or at least *a* standard) markup language for Web pages relating to mixed drinks. HHML-oriented browser extensions or plug-ins might emerge. People would expect drink-related sites to expose at least part of their content using standard HHML tags. Vendors of drink ingredients would want to make recommended drink

[26] *See id.* at 177.

recipes on their sites, based on their ingredients, accessible to HHML-enabled browsers.

Or so we can imagine, for illustrative purposes. There is no HHML, and with any luck there never will be, but other XML schemas are already in use, both on the Web and in other forms of information interchange. They include, among others, schemas for distribution of news stories, automotive information, and financial, insurance, and medical data.[27] The predicted XML-based transformation of the Web depends on the proliferation of such XML schemas. If an XML schema became prevalent in the way hypothesized here, the commercial implications of copyright ownership in it could be substantial, at least in its particular zone of application. The copyright owner could not adopt too strict an enforcement policy at the outset, or no one would use the markup language based on that schema. Commercial success ordinarily would depend on widespread use. But if the markup language did become a standard, copyright ownership in it could be used selectively to protect the copyright owner's technological and market position.

This point was first noticed in the software industry context more than a decade ago, when Ashton-Tate Corporation, then the publisher of dBASE database management software, sued competing software companies for allegedly infringing Ashton-Tate's asserted copyright in the dBASE language by supplying compatible database software.[28] The compatible software was built with different code but allowed users to employ the same dBASE language available in dBASE itself. In response to the lawsuit, the main defendant, Fox Software, Inc., sent an "[o]pen [l]etter to [the dBASE] [l]anguage [c]ommunity."[29] The open letter stated that Ashton-Tate and its founder, George Tate,[30] had "encouraged all of us—not just Fox—to produce applications, compilers, interpreters, report writers, program generators, etc. which utilize and support the [dBASE] language," to maximize Ashton-

[27] TRAVIS, *supra* note 2, at 22, 31-32.

[28] Elizabeth G. Lowry, Comment, *Copyright Protection for Computer Languages: Creative Incentive or Technological Threat?*, 39 EMORY L.J. 1293, 1294 (1990).

[29] *Fox Software/Ashton-Tate: Fox Software Open Letter to [dBASE] Language Community Regarding Fox Software and Ashton-Tate Litigation*, BUS. WIRE, Dec. 13, 1988 LEXIS, Nexis Library, News Group File, All, *quoted in* Lowry, *supra* note 28, at 1310 n.73.

[30] "Ashton" was Tate's pet parrot. *See* Victor F. Zonana, *Ashton-Tate; Confronting a Hard Life in the World of Software*, L.A. TIMES, May 10, 1987, LEXIS, Nexis Library, News Group File, All.

Tate's market share.[31] According to the letter, when Ashton-Tate sued Fox, Ashton-Tate's share had reached sixty percent of the market.[32]

Although the parties settled the Ashton-Tate litigation before any decision on the language-copyrightability issue,[33] the dBASE example shows that vigorous efforts to promote standardization on a particular language do not preclude eventual efforts by the author of that language to use copyright against actual or potential competitors for the supply of software or associated tools or services. Thus, the dependence of XML-based markup languages on wide acceptance does not ensure that they will be placed in the public domain. Whether copyright law will support copyright claims in XML-based markup languages is, accordingly, a question worth exploring—both because of its implications for copyright doctrine, and in its own right.

III. COPYRIGHTABILITY OF COMPUTER LANGUAGES

Whether computer languages in general are capable of being protected by copyright is an "unsettled and confusing area of law."[34] No reported court decision expressly determines this issue.[35] Ashton-Tate sued its competitors in personal computing antiquity (1988), but the case produced no relevant precedent.[36] During medieval times (the early and mid-1990's), Lotus Development Corporation pursued litigation against Borland International, Inc., for infringing Lotus's alleged copyright in the Lotus 1-2-3 "menu command hierarchy," but the litigation eventually was resolved without

[31] *Fox Software/Ashton-Tate: Fox Software Open Letter to [dBASE] Language Community Regarding Fox Software and Ashton-Tate Litigation*, *supra* note 29.

[32] *Id.*

[33] *See* Richard O'Reilly, *The Future of Database Programs*, L.A. TIMES, Apr. 23, 1992, at D-3, LEXIS, Nexis Library, News Group File, All.

[34] Michael P. Doerr, Note, *Java: An Innovation in Software Development and a Dilemma in Copyright Law*, 7 J. INTELL. PROP. L. 127, 127 (1999).

[35] *Id.* In *Sun Microsystems, Inc. v. Microsoft Corp.*, 188 F.3d 1115, 51 U.S.P.Q.2d (BNA) 1825 (9th Cir. 1999), Sun alleged that Microsoft had committed copyright infringement by exceeding the scope of a license agreement under which Sun had licensed the Java programming language to Microsoft. *Id.* at 1117. The court did not address the copyrightability of Java, *see id.*, and the parties subsequently settled the dispute. Linda Leung, *Client; Sun and Microsoft Settle Their Java Licensing Row*, COMPUTING, Feb. 1, 2001, at 28 LEXIS, Nexis Library, News Group File, All.

[36] *See* O'Reilly, *supra* note 33, at D-3.

determining the copyrightability of computer languages.[37] This key question of law thus remains without an authoritative answer.

Computer languages may well embody patentable inventions if they possess sufficient novelty and non-obviousness.[38] Copyright protection, however, if it subsists at all, automatically applies without the formalities, costs, and delays of patent prosecution or the burdens of establishing patentability.[39] The automatic nature of copyright, and the absence of any need to show novelty or non-obviousness, make copyright a vital asset in the information technology marketplace.

Under the Copyright Act, "[c]opyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression . . ." as soon as they are so fixed.[40] Protection for an original work of authorship does not, however, "extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."[41]

The dichotomy between copyrightable "expression" and uncopyrightable "ideas," "procedures," and the like codified in the Copyright Act was set forth in *Baker v. Selden*, 101 U.S. 99 (1880), in which the Supreme Court, more than a century ago, rejected a claim of copyright in an accounting system described in a book.[42] The Court declared: "The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself."[43] Protection for the art itself would arise, if at all, from issuance of a patent, not as a matter of copyright.[44]

[37] Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807, 815, 34 U.S.P.Q.2d (BNA) 1014, 1021 (1st Cir. 1995), *aff'd by an equally divided Court*, 516 U.S. 233 (1996). An earlier Lotus action against Paperback Software, International, produced a favorable result for Lotus, but the district court rejected an argument based on the asserted uncopyrightability of computer languages without clearly addressing whether computer languages are copyrightable subject matter. *See* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 73, 15 U.S.P.Q.2d 1577, 1604 (D. Mass. 1990).

[38] *See* Marci A. Hamilton & Ted Sabety, *Computer Science Concepts in Copyright Cases: The Path to a Coherent Law*, 10 HARV. J. LAW & TECH. 239, 263 n.95 (1997).

[39] *See* 17 U.S.C. § 102(a) (2001).

[40] *Id.* In *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991), 18 U.S.P.Q.2d (BNA) 1275 (1991), the Supreme Court held that works of authorship are "original" unless they fall into the "narrow category of works in which the creative spark is utterly lacking or so trivial as to be virtually nonexistent." *Id.* at 359. The Court stated: "As a constitutional matter, copyright protects only those constituent elements of a work that possess more than a *de minimis* quantum of creativity." *Id.* at 363.

[41] 17 U.S.C. § 102(b).

[42] Baker v. Seldon, 101 U.S. 99 (1880).

[43] *Id.* at 105.

[44] *Id.* at 102-03.

In assessing copyrightability of languages, one may distinguish between a specification or other description of the language (potentially analogous to the book in *Baker*) and the language itself (potentially analogous to the "art" that the book describes). Just as Danish is broader and richer than any Danish dictionary or Danish grammar, a computer language is more than, and different from, any of the myriad books that may have been written about it. Few are likely to dispute that a language specification or other description of the language is copyrightable. Likewise, a computer program written in a computer language is protectable by copyright.[45] The issue, at least as ordinarily viewed, is whether the author of an original language specification owns a copyright in the language itself, so that anyone who uses the language—*i.e.*, creates sentences or writes programs in the language, enables others to do so, or modifies the language—without the author's permission is a copyright infringer.

Computer languages, unlike most human languages, are created by individuals and thus, potentially, are "works of authorship." Although new computer languages may draw on concepts from prior languages, they are often likely to contain original aspects.[46] For that matter, the specification of a computer language may be a supremely creative original work of authorship.[47]

Learned Hand seemed almost to anticipate the issue faced today. In *Reiss v. National Quotation Bureau, Inc.*, Hand concluded that a telegraph codebook containing 6,325 coined, meaningless words was copyrightable.[48]

[45] *See* 17 U.S.C. § 101; Hamilton & Sabety, *supra* note 38, at 240.

[46] The originality requirement for copyright requires "little creativity." Doerr, *supra* note 34, at 141. Thus, "a computer language in all likelihood meets the originality standard," although "its developer may have difficulty establishing which elements of the language are original and which are copied from a preexisting language." Lowry, *supra* note 28, at 1308, *quoted in* Doerr, *supra* note 34, at 141.

[47] *Cf.* Ronald L. Johnston & Allen R. Grogan, *Copyright Protection for Command Driven Interfaces*, 8 COMPUTER LAW. 1, 2 (June 1991) (emphasizing the creativity required to develop a user interface, including predefined command sequences).

[48] 276 F. 717, 718 (S.D.N.Y. 1921). Kenneth W. Dam, *Some Economic Considerations in the Intellectual Property Protection of Software*, 24 J. LEGAL STUD. 321, 324 n.14 (1995) (providing factual background).

> The development of telegraphy led to the preparation of code books, consisting of lists of 'words' that had no ordinary meaning, which might appear alone or accompanied by another list of common expressions within a particular area of commerce. By assigning meanings to the code words, persons could communicate secret messages to one another. The copyrightability of such compilations was accepted without difficulty, even though they were manifestly entirely functional and the lists of code words communicated nothing whatever on their own.

This was the context in which Hand wrote of a "blank Esperanto."[49] He added: "Suppose a mathematician were to devise a new set of compressed and more abstract symbols, and left them for some conventional meaning to be filled in."[50] According to Hand, these sets of symbols, to which the codebook was similar, would be copyrightable, and the codebook therefore was as well.[51] But Hand's analysis also contains blanks.[52]

Indeed, despite such early decisions upholding copyright in telegraph codebooks,[53] most law review commentary has rejected the idea that computer languages are subject matter in which copyright protection can subsist.[54] Ten years ago, one commentator wrote:

> Until quite recently few observers would have considered copyright protection for computer programming language to be a matter of legal controversy, or even concern. The general assumption was that computer programming languages were not subject to copyright protection because they were unprotectable "ideas," rather than protectable "expressions" of ideas.[55]

The issue of computer language copyrightability has become an explicit question, rather than a matter of assumption, as the software industry has evolved and software companies have asserted copyright protection in menu

Lloyd L. Weinreb, *Copyright for Functional Expression*, 111 HARV. L. REV. 1149, 1192 (1998).

[49] *Reiss*, 276 F. at 718.

[50] *Id.*

[51] *Id.* at 718, 719.

[52] After speaking of mathematical symbols to which meaning had not yet been assigned, Hand dismissed *Baker v. Selden* as "too foreign to the case at bar to deserve comment," *id.* at 719, without explaining why. This omission calls into question Hand's conclusion. In *Baker*, the Supreme Court specifically stated: "The copyright of a work on mathematical science cannot give to the author an exclusive right to the methods of operation which he propounds, or to the diagrams which he employs to explain them, so as to prevent an engineer from using them whenever occasion requires." 101 U.S. at 103.

[53] Hartfield v. Peterson, 91 F.2d 998, 1000 (2d Cir. 1937) (Augustus Hand, J.); *Reiss*, 276 F. at 719.

[54] *See, e.g.*, Hamilton & Sabety, *supra* note 38, at 242; Lowry, *supra* note 28, at 1296; Richard H. Stern, *Copyright in Computer Programming Languages*, 17 RUTGERS COMPUTER & TECH. L.J. 321, 334 (1991); Steve Posner, *Can a Computer Language Be Copyrighted? The State of Confusion in Computer Copyright Law*, 11 COMPUTER/L.J. 97, 97 (1991).

[55] Stern, *supra* note 54, at 322.

command hierarchies,[56] graphical user interfaces,[57] and new languages such as Java.[58]

In addition to various policy arguments,[59] two primary arguments based on copyright law principles have been advanced against recognizing copyright protection for computer languages themselves, as opposed to the specifications that define them or the code that implements them on a particular machine. The first such argument is that computer languages are not protected by copyright because, under *Baker v. Selden* and the modern Copyright Act, they are uncopyrightable "systems" or other "ideas."[60] The second is that languages are not protected because they are not fully fixed in a tangible medium of expression.[61] These arguments, not yet tested in any reported decision, raise serious questions that must be addressed by an assessment of possible copyright protection for computer languages.

A. COMPUTER LANGUAGES AS SYSTEMS

The most straightforward argument against copyrightability is that "computer languages are systems or ideas."[62] The Copyright Act makes clear that, even though a system may be embodied or explained in an original work of authorship, the system itself is not copyrightable.[63] "Like any language, a computer language is a *system* of vocabulary and grammar

[56] *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807, 809 (1st Cir. 1995), *aff'd by an equally divided Court*, 516 U.S. 233 (1996).

[57] *See* Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435, 1438 (9th Cir. 1994), *cert. denied*, 513 U.S. 1184 (1995).

[58] *See* Doerr, *supra* note 34, at 127.

[59] *See* Posner, *supra* note 54, at 100-03.

[60] *See, e.g.*, Stern, *supra* note 54, at 353; Lowry, *supra* note 28, at 1296.

[61] Hamilton & Sabety, *supra* note 38, at 269. Hamilton and Sabety also contend that computer language copyright also violates "First Amendment principles." *Id.* at 270. They state: "By authorizing protection for languages, the Act would be authorizing prior restraint of any expression in that language." *Id.* The authority they cite, however, merely reaffirms that "[n]o author may copyright his ideas or the facts he narrates." Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 556, 225 U.S.P.Q. (BNA) 1073, 1079 (1985), *quoted in* Hamilton & Sabety, *supra* note 38, at 270 n.119. The question thus reduces to the first of the two issues highlighted in the text, namely whether a computer language constitutes an unprotectable "idea" or "system" rather than protectable "expression." Although the uncopyrightability of ideas is important in making copyright consistent with First Amendment principles, one need not invoke separate First Amendment doctrine to resolve the language copyright issue.

[62] Lowry, *supra* note 28, at 1296.

[63] 17 U.S.C. § 102(b) (1994).

rules."[64] Therefore, according to this argument, the language is not copyrightable.

Indeed, a leading dictionary includes a definition of "language" as "a formal *system* of signs and symbols (as *FORTRAN* [a computer language] or a calculus in logic) including rules for the formation and transformation of admissible expressions."[65] The plain terms of the Copyright Act, coupled with this established definition of "language" in the computer context, can be cited in support of the view that, no matter how original, computer languages are not a proper subject of copyright protection.

In *Lotus Development Corp. v. Borland International, Inc.*,[66] the First Circuit, although not specifically addressing language copyrightability, took a similar approach to determining the copyrightability of the menu command hierarchy of the Lotus 1-2-3 spreadsheet software. The court held that the command hierarchy, which also provided key elements for use in the Lotus macro language, was an uncopyrightable "method of operation."[67] If a menu command structure falls within the plain meaning of "method of operation," as used in the statute, a language may well fall within the plain meaning of "system."

The Supreme Court, however, was equally divided on the First Circuit's approach.[68] It remains similarly uncertain how the "system" approach would be resolved. The word "system" is open to multiple potential meanings. A software application that is not in any sense a language, and that is widely acknowledged to be copyrightable, may also be characterized as a "system." For example, the term "accounting system," which describes the idea held uncopyrightable in *Baker v. Selden*,[69] may equally well, in modern parlance, refer to a software package[70] that all would agree is protected by copyright. As discussed later in this Article, resolving the question in the XML context

[64] Lowry, *supra* note 28, at 1312 (emphasis added).

[65] MERRIAM-WEBSTER'S COLLEGIATE DICTIONARY, *at* http://www.britannica.com/dictionary (last visited Aug. 30, 2001) (emphasis added).

[66] 49 F.3d 807, 34 U.S.P.Q.2d (BNA) 1014 (1st Cir. 1995), *aff'd by an equally divided Court*, 516 U.S. 233 (1996).

[67] 49 F.3d at 815.

[68] *See* 516 U.S. at 233.

[69] *See* Am. Dental Ass'n v. Delta Dental Plans Ass'n, 126 F.3d 977, 981, 44 U.S.P.Q.2d (BNA) 1296, 1300 (7th Cir. 1997) (referring to Selden's "accounting system").

[70] *See* Coffey v. Dobbs Int'l Servs., Inc., 170 F.3d 323, 325 (2d Cir. 1999) (referring to "a computer software billing and accounting system").

depends on defining the relevant "system" or idea and considering how it corresponds to the expressive aspects of the XML schema.

B. COMPUTER LANGUAGES AS NOT FULLY FIXED

A related but distinct argument against copyrightability of computer languages, advanced by Marci Hamilton and Ted Sabety, is that holding languages copyrightable would provide copyright protection for expressions not yet fixed in a tangible medium of expression, contrary to the Copyright Act's express requirement.[71] For that matter, one could say that language copyright would treat as copyrighted any potential expression in the language, even though it has not yet occurred in any medium, tangible or otherwise.

Hamilton and Sabety point out that computer language copyright could apply to one of two files: "a list of all possible sentences in that language, or an expression of its specification . . . that fully describes the language."[72] The first is "likely to be impossible," because all possible sentences in any but the most simplistic language have not yet been uttered, and the second is "problematic," because the language specification "is not fixation of the sentence itself."[73]

Michael Doerr criticizes this argument against copyrightability because it "presupposes that in order for a programming language to be fixed, all of the possible sentences that could be formed using the language must also be fixed."[74] Instead, according to Doerr, "if one wished only to obtain copyright protection for the language itself [as opposed to all possible sentences], one should be able to fulfill the fixation requirement by simply fixing the language and the grammar rules for the language in a computer program utilizing the language."[75]

The problem with this opposing view is that it merely shifts the question from one about fixation to one about infringement. If the language specification is the original work of authorship that is fixed in a tangible medium of expression, is that copyright infringed when someone does not

[71] Hamilton & Sabety, *supra* note 38, at 269.
[72] *Id.*
[73] *Id.*
[74] Doerr, *supra* note 34, at 140.
[75] *Id.* at 141.

directly copy the language specification, but merely *uses* it to create sentences that have *not* previously been fixed?

The author of a language specification could assert that unauthorized use of the language specification constitutes unauthorized reproduction or unauthorized creation of derivative works.[76] In either instance, there would need to be actual unauthorized copying of at least some portion of the language specification. At a minimum, "[a] derivative work must *incorporate* a protected work in some concrete or permanent 'form.' "[77] To *use* a specification is not necessarily to *copy* or *incorporate* it.

For copying of a copyrighted work to be legally actionable, there must be not only copying but "substantial similarity between the two works."[78] This requirement has also been applied to finding unauthorized creation of derivative works.[79] Determining whether similarity is substantial "presents one of the most difficult questions in copyright law, and one that is the least susceptible of helpful generalizations."[80]

In *Vault Corp. v. Quaid Software, Ltd.*,[81] the Fifth Circuit addressed a contention that Vault had created an unauthorized derivative work when it copied about thirty characters of Vault's software locking program into Quaid's, eighty-page lock-cracking program.[82] Pointing out that the two programs served different purposes, the court held that they were not substantially similar.[83] The difference in purpose between a language specification and sentences written in the language is likely to be even greater

[76] 17 U.S.C. §§ 106(1)(2) (1994) (stating that the rights to reproduce and prepare derivative works are the exclusive rights of the copyright owner). Anyone who violates any of the copyright owner's exclusive rights is an infringer. 17 U.S.C. § 501(a) (1994).

[77] Lewis Galoob Toys, Inc. v. Nintendo of America, Inc., 1992 U.S. App. LEXIS 17680, *3 (9th Cir. Aug. 5, 1992) (emphasis added). "The examples of derivative works provided by the [Copyright] Act all *physically incorporate* the underlying work or works. The Act's legislative history similarly indicates that 'the infringing work must incorporate a portion of the copyrighted work in some form.' " *Id.* at *4 (citing 1976 U.S. CODE CONG. & ADMIN. NEWS 5659, 5675) (emphasis added). *Accord* Ets-Hokin v. Skyy Spirits, Inc., 225 F.3d 1068, 1079 (9th Cir. 2000).

[78] Eng'g Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1341 (5th Cir. 1994), *supplemented on other grounds*, 46 F.3d 408 (5th Cir. 1995). *Accord* 4 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[A] (2000) [hereinafter NIMMER].

[79] *See* Michael Wurzer, Note, *Infringement of the Exclusive Right to Prepare Derivative Works: Reducing Uncertainty*, 73 MINN. L. REV. 1521, 1521 (1989).

[80] 4 NIMMER, *supra* note 78, § 13.03[A] at 13-27 (footnote omitted).

[81] 847 F.2d 255 (5th Cir. 1988).

[82] *Id.* at 267-68.

[83] *Id.* at 268.

than the difference between two programs. A finding of substantial similarity, at least under the *Vault* approach, thus seems unlikely.

This conclusion is reinforced by the more recent decision of the Ninth Circuit in *Apple Computer, Inc. v. Microsoft Corp.*[84] The court's decision in *Apple* reflects the principle that "more similarity is required when less protectible matter is at issue."[85] Taking into account "the limited number of ways that the basic ideas of the Apple [graphical user interface] can be expressed differently," as well as a license covering a large part of the interface, the court held: "[W]e conclude that only 'thin' protection, against virtually identical copying, is appropriate."[86] Insofar as computer languages tend to be "idea"-intensive, at the "thin" end of the protectability spectrum, the principle applied in *Apple* further suggests that sentences written in a language are unlikely to be considered unauthorized copies or derivative works.

Accordingly, even if a computer language specification is fixed, uses of the language to create previously unfixed sentences may not infringe the copyright in the specification itself. So far, however, we have considered the question in the abstract. Actual practice is more complicated and rapidly changing. Some programming languages, for example, are provided in software packages that include, and perhaps require, run-time interpreters and actual code modules.[87] Unlicensed copying of such interpreters or modules, if they are copyrighted, is infringing. Thus, even if a language itself in principle is uncopyrightable, practical use of it may involve copying of copyrighted materials. In the next section, we consider language copyrightability in the specific context of XML-based markup languages.

IV. THE SPECIAL CASE OF XML SCHEMAS

According to W3C's Director, "XML makes it easy for everyone to create their own tags or entire markup languages."[88] Nevertheless, XML schemas are not entirely self-contained, and they are not necessarily "languages" in a

[84] 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 513 U.S. 1184 (1995).

[85] 4 NIMMER, *supra* note 78, § 13.03[A] at 13-28.

[86] 35 F.3d at 1442.

[87] *See* Brett Frischmann & Dan Moylan, *The Evolving Common Law Doctrine of Copyright Misuse: A Unified Theory and Its Application to Software*, 15 BERKELEY TECH. L.J. 865, 905-06 (2000) (describing compiled and interpreted programs).

[88] BERNERS-LEE, *supra* note 10, at 161.

formal sense. Rather, XML schemas are XML extensions: XML schemas are used in or with XML documents; an XML parser[89] is required to process an XML document; and the XML document must substantially conform to the rules of XML, including the rules that make it possible for various schemas, with potentially conflicting vocabularies, to coexist and successfully interact on the Web.[90]

Although XML schemas are often called markup languages, they are also sometimes referred to as "dialects"[91] or "vocabularies"[92] of XML or as XML "applications."[93] "By definition, a language is a formal system consisting of three elements: a vocabulary, a syntax, and semantics."[94] XML supplies basic signs and symbols (a partial vocabulary) that interrelate based on a set of rules (syntax), but leaves meaning (semantics) largely to the schema author. The XML schema, in turn, has no syntax of its own, but the schema uses XML's syntax to define a set of markup tags (an extended vocabulary) and to give them meaning, which may include defining acceptable contents and values for particular elements and attributes[95]—hence, the "*Semantic* Web."[96]

We, thus, may refine the description of XML schemas as "markup languages" to say that, in a formal sense, the markup language consists not of the schema alone but of XML as extended by the schema. Documents written in this language, in turn, must be interpreted by an XML parser that has access to the literal terms of the schema. These characteristics of XML schemas raise several questions. First, does the dependence of the XML schema on XML, a standard of W3C, affect schema "copyrightability" questions? Second, focusing on the XML schema as potentially copyrightable in its own right, does the schema's use of syntax from a source

[89] "Any computer program that is supposed to generate or accept sentences in a computer language requires some kind of parser." Hamilton & Sabety, *supra* note 38, at 267. The parser "accepts strings of symbols consistent with the grammar [of a computer language] and rejects those that are grammatically incorrect." *Id.* at 266.

[90] *See* BERNERS-LEE, *supra* note 10, at 161-62.

[91] "XML . . . makes it possible for developers to create their own mutually interoperable dialects of markup languages, including but not limited to HTML." ST. LAURENT, *supra* note 3, at 1.

[92] TRAVIS, *supra* note 2, at 51.

[93] CASTRO, *supra* note 6, at 13.

[94] Pamela Samuelson, *Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of Lotus v. Paperback*, 55 LAW & CONTEMP. PROBS. 311, 334 (Spring 1992). *Accord* Stern, *supra* note 54, at 327.

[95] *See* CASTRO, *supra* note 6, at 22.

[96] *See* Berners-Lee et al., *supra* note 5, at 36.

external to the schema affect the analysis? Third, does the need to load a literal copy of the XML schema into a computer's memory make a difference?

A. XML AND W3C

The current specification for XML, as of this writing, is set forth in a W3C Recommendation, dated October 6, 2000, entitled "Extensible Markup Language (XML) 1.0 (Second Edition)."[97] The specification bears a copyright notice and states that W3C "software licensing rules apply."[98] Thus, although XML is an open standard, W3C apparently maintains that at least the specification itself is copyrightable.

The W3C software licensing rules state, however, that "[p]ermission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted," provided only that the user includes a specified W3C copyright notice.[99] The W3C rules do not say that the licensee must place any resulting modifications or works created with XML in the public domain or make them available to the public on comparable license terms. Subject to the proviso, the "software" may be used, copied, modified, and distributed "for any purpose," which includes any commercial purpose.

W3C could have crafted its license to state that XML, or at least the XML specification, could be used only in conjunction with schemas that were in the public domain or otherwise freely available. This "viral" model of public licensing is exemplified by the GNU General Public License (GPL).[100] Under Section 2(a) of the GPL, anyone can modify a work that is licensed to the public under the GPL, but the person modifying the work must agree to make the modified work available "to all third parties" under the terms of the GPL.[101] Because W3C does not follow the viral licensing model of the

[97] *At* http://www.w3.org/TR/2000/REC-xml-20001006 (last visited Aug. 30, 2001).

[98] *Id.*

[99] *W3C Software Notice and License, at* http://www.w3.org/Consortium/Legal/copyright-software-19980720 (last visited Aug. 30, 2001).

[100] *At* http://www.gnu.org/copyleft/gpl.html. "GNU is a recursive acronym for 'GNU's Not Unix'; it is pronounced 'guh-NEW.' " *GNU Home Page, at* http://www.gnu. org/ (last visited Aug. 30, 2001).

[101] *GNU General Public License,* Version 2, *at* http://www.gnu.org/copyleft/gpl.html (June 1991) (last visited Aug. 30, 2001). An interesting question, beyond this Article's scope, is whether this term is enforceable. Licensing terms that seek to leverage the copyright owner's rights under copyright law to create rights not granted by copyright law may be challenged on grounds of copyright misuse. *See*

GPL with respect to XML, W3C's assertion of copyright in the XML specification appears to have few implications for copyrightability of XML schemas.[102]

B. USE OF XML SYNTAX

That XML schemas use XML syntax, rather than syntax created by the schema author, appears unlikely to affect schema copyrightability. Although this fact makes XML schemas less than complete "languages," the key for determining copyrightability, in this context, is whether the proposed subject matter of copyright is a "system" or similarly unprotectable idea. Nothing in the concept of a "system," either as used in the Copyright Act or as the term is generally understood, excludes systems that are only "subsystems." On the contrary, Section 102(b) of the Copyright Act uses the word "system" along with words such as "procedure" and "process," neither of which suggests completeness.

To be sure, an XML schema may define one tag or one hundred. Whether to call it a "system," or even a "procedure" or "process," may depend on it possessing more than *de minimis* content. But a schema with *de minimis* content might not qualify for copyright protection anyway, because it might lack sufficient originality. Single words and short phrases are not copyrightable.[103] That XML schemas are something less than complete languages does not establish that they are not ideas.

Lasercomb Am., Inc. v. Reynolds, 911 F.2d 970, 977, 15 U.S.P.Q.2d (BNA) 1846, 1852 (4th Cir. 1990) (stating that public policy forbids the use of copyright "to secure an exclusive right . . . not granted" by copyright); *accord* Practice Mgt. Info. Corp. v. AMA, 121 F.3d 516, 520, 45 U.S.P.Q.2d (BNA) 1780 (9th Cir. 1997), *cert. denied*, 522 U.S. 933 (1997), *modified on other grounds*, 133 F.3d 1140 (9th Cir. 1998), *cert. denied*, 524 U.S. 952 (1998).

[102] W3C states that its software licensing terms are designed to be "no more restrictive than GPL," *W3C Software Notice and License*, *supra* note 99, but they lack GPL's viral aspect. Conceivably W3C could assert that XML schemas lacking the specified W3C copyright notice were unauthorized derivative works, but such claims seem unlikely from a practical standpoint given W3C's apparent desire to maintain XML as an open standard.

[103] *See* Arica Inst., Inc. v. Palmer, 970 F.2d 1067, 1072, 23 U.S.P.Q.2d (BNA) 1593, 1597 (2d Cir. 1992)("[S]ingle words or short phrases . . . do not exhibit the minimal creativity required for copyright protection."); 37 C.F.R. § 202.1(a) (prohibiting copyright registration of "words and short phrases such as names, titles and slogans").

C. COPYING OF XML SCHEMAS

Files created in any programming language (other than machine language) either must be compiled and linked to create object code or they must be used in conjunction with an interpreter.[104] Much of the discussion in the existing literature about language copyright assumes, at least implicitly, that the issue is whether one person can create a compiler or interpreter, using different code, for a language that another person claims to own.[105] The compiler or interpreter is understood to be copyrightable.[106] If the language is not copyrightable, however, competing compilers and interpreters for that language (based on different code) may be created and used without infringement.

XML markup languages present a different situation. An XML schema together with XML defines a markup language in which XML documents can be written. For an XML parser to interpret such documents, however, the parser must have full access to the schema. Thus, a file setting forth the schema must be copied into the memory of a computer at some step in the process. XML schemas may be stored on the same system as the XML document (in the same file or a separate file) or on "public" servers.[107] Either way, however, they need to be loaded into memory, and thus copied,[108] to make the XML document ultimately readable.

The parser's need for access to the XML schema as part of, or together with, the XML document makes the second argument against language copyrightability—that copyright protection for a language would protect unfixed expressions, or (as suggested above) that use of a language specification is not infringement—insufficient to dispose of copyrightability in the XML schema context. The reason is that the user of an XML schema is not merely writing sentences in the language and using an independently-created compiler or interpreter, which is likely to contain copies of only relatively small portions of the expression that is contained in the language specifica-

[104] *See* Frischmann & Moylan, *supra* note 87, at 905-06 (explaining methods of translation of source code into object code).

[105] *See, e.g.*, Doerr, *supra* note 34, at 133; Lowry, *supra* note 28, at 1339.

[106] *See* Lowry, *supra* note 28, at 1303 (explaining that computer programs that translate high-level computer language into lower-level language are copyrightable).

[107] *See* CASTRO, *supra* note 6, at 40.

[108] *See* MAI Sys. Corp. v. Peak Computer, Inc., 991 F.2d 511, 518, 26 U.S.P.Q.2d (BNA) 1458, 1465 (9th Cir. 1993) (copying a computer file into memory is potentially infringing).

tion. Rather, the user of the XML schema is writing sentences that can only create a usable XML document if the XML schema itself, which is fully fixed, is also fully copied.

Although one could try to write a different but "compatible" schema file for a given XML-based markup language, this file is quite likely to be substantially similar to the original schema file. This is so because the schema—unlike a compiler or an interpreter—is little more than a functional language specification, and because an XML schema is sharply constrained by XML itself. If the two files were actually to define, in a mutually compatible manner, the same schema, they could differ only in immaterial, arbitrary ways, such as by changing the order of some of their contents.

For all these reasons, use of someone else's XML schema ordinarily will involve copying a fixed work (the schema) in a way, and to an extent, that use of someone else's programming language with an independently-developed compiler or interpreter does not. That is, an independently-developed compiler or interpreter, although perhaps copying isolated elements of a language specification, need not contain extensive copying of the language specification or of any other compiler or interpreter for that language. Using an independently-developed compiler or interpreter, one can execute a program written in the language without ever substantially copying someone else's work. An XML parser cannot, however, process an XML document written in the markup language defined by a particular schema without fully copying the schema—someone's work of authorship—into a computer's memory.

Hamilton and Sabety's point about fixation, or the related point about infringement noted above, is apt when applied to use of an independently-developed, non-infringing compiler or interpreter. The fixation point is less relevant in the XML schema context because the infringement argument does not depend on showing that sentences expressed using the language are infringing. Rather, there is likely to be substantial literal copying, otherwise sufficient to give rise to infringement, of the XML schema itself when it is loaded along with the XML document.[109] If the XML schema is

[109] We may assume, for purposes of this discussion, that the schema is sufficiently original to be an otherwise protectable expression. If the schema is derived from common terms or merely adapts a pre-existing data model, there may be a question of originality. The current focus, however, is on whether an XML schema that otherwise constitutes an original work of authorship is nevertheless unprotectable as an uncopyrightable idea.

uncopyrightable, then the basis for this conclusion most likely will be found not in the fixation argument, but in the schema's nature as a "system" or similarly uncopyrightable idea. Such a finding needs to show that, when it comes to XML, the schema *is* the system.

V. THE WORK AND THE IDEA

Under Section 102(b) of the Copyright Act, the "idea" expressed by a work of authorship is uncopyrightable "regardless of the form in which it is described, explained, illustrated, or embodied in [an otherwise protected] work."[110] Copyright law further provides: "[W]hen an idea can be expressed in very few ways, copyright law does not protect that expression, because doing so would confer a *de facto* monopoly over the idea. In such cases idea and expression are said to be merged."[111] The uncopyrightability of the idea thus trumps the copyrightability of the expression.

Like the similarly broad and foundational doctrine of fair use, the principle that ideas are not copyrightable under United States law is traceable to a line of decisional authority that began well over a century ago.[112] The uncopyrightability of ideas, even when they are embodied in otherwise copyrightable works of authorship, is "[p]art of the information ethos in the United States that facts and ideas cannot be owned, suppressed, is censored, or regulated . . . ."[113] This doctrine is basic not only to the ability to express ideas, but also to the ability to read and otherwise receive them.[114] The doctrine thus goes to the heart of the constitutional purpose of copyright,

[110] 17 U.S.C. § 102(b) (1994). The same is true of any "procedure, process, system, method of operation, concept, principle, or discovery" embodied in the work. *Id.*

[111] Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527, 533, 29 U.S.P.Q.2d (BNA) 1747, 1750 (5th Cir. 1994). *Accord* Kregos v. Associated Press, 937 F.2d 700, 705, 19 U.S.P.Q.2d (BNA) 1161, 1163 (2d Cir. 1991) ("The fundamental copyright principle that only the expression of an idea and not the idea itself is protectable has produced a corollary maxim that *even expression is not protected* in those instances where there is only one or so few ways of expressing an idea that protection of the expression would effectively accord protection to the idea itself.") (emphasis added). *But see* Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435, 1444, 32 U.S.P.Q.2d 1086, 1093 (9th Cir. 1994) ("When an idea and its expression are indistinguishable, or 'merged,' the expression will only be *protected against nearly identical copying.*") (emphasis added), *cert. denied*, 513 U.S. 1184 (1995).

[112] *See, e.g.*, Baker v. Selden, 101 U.S. 99 (1879) (originating the idea-expression dichotomy); Folsom v. Marsh, 9 F. Cas. 342 (C.C.D. Mass. 1841) (originating the fair use doctrine).

[113] JESSICA LITMAN, DIGITAL COPYRIGHT 11 (2001).

[114] *See id.*

which is "[t]o promote the Progress of Science,"[115] and likewise enables copyright to coexist with the First Amendment's guarantee of free expression.[116] Moreover, it gives effect to the constitutional and statutory distinction between copyright and patent, as well as the legislative scheme for establishing patentability.[117]

The idea-expression dichotomy, and the merger doctrine that flows from it, are sufficiently fundamental to copyright law in the United States that they have been largely immune to the tendency in copyright legislation of the last century to adopt more detailed statutory provisions that reflect political compromises among existing copyright industries.[118] Because the merger doctrine has not been defined in detailed provisions authored or influenced by particular copyright interests, the courts retain a relatively broad compass to apply the doctrine to new technology.

Courts have held that the merger doctrine is "particularly applicable" to subject matter involving abstract rules, such as games and contests.[119] On this basis, for example, the Ninth Circuit held that the merger doctrine applied to "the expression of the rules of . . . game manuals" when the author had been "unable to distinguish the expression of the rules from the idea of the rules themselves."[120] The merger doctrine also played a key role in denying copyright protection to Apple Computer for significant aspects of its graphical user interface.[121] Because, as outlined above, use of an XML schema requires substantial literal copying of expression, the merger doctrine is pivotal here as well.[122]

[115] U.S. CONST. art. I, § 8, cl. 8.

[116] U.S. CONST. amend. I; *see* Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 556 (1984).

[117] *See Baker*, 101 U.S. at 102-03.

[118] *See generally* LITMAN, *supra* note 113, at 35-69.

[119] Allen v. Academic Games League of Am., Inc., 89 F.3d 614, 617 (9th Cir. 1996).

[120] *Id.*

[121] *See Apple Computer*, 35 F.3d at 1444.

[122] Edward Samuels suggests that "[m]any of the traditional examples of [unprotected ideas, systems, and the like] are actually more properly viewed as examples of the works of utility doctrine . . . ." EDWARD SAMUELS, THE ILLUSTRATED STORY OF COPYRIGHT 188 (2000). Samuels states:

> [A] treatise on relativity or perspective or accounting creates rights in the words used to explain the principles, but doesn't give any exclusive rights in the use of the principles themselves. These examples could all be explained either because the principles are to some extent "functional," and therefore not protected under the works of utility doctrine, or because they are "ideas," separate and apart from any description of the principles.

Id. at 189. The works of utility doctrine plainly does not, however, preclude copyright protection for

Does an XML schema embody a language system, subsystem, or other unprotected category, such as a procedure or process, that copyright law treats as an "idea" for purposes of the idea-expression dichotomy? If so, does this system "merge" with the expression of the schema, so as to render the schema uncopyrightable, even though it contains original expression that otherwise would be protectable? The answers depend in significant part on how the idea associated with the schema is defined. No one schema is the only way to express the idea, generally, of an XML-based markup language. Nor is any one likely to be the only way to express the idea of an XML-based markup language relating to the general subject matter of that schema (such as the imaginary markup language outlined earlier relating to mixed drinks). If the idea is defined in such broad terms, merger will not occur.

If the idea is defined more narrowly, however, the result may well be different. This Article suggests that, at least for a functional work like an XML schema, an approach more compatible with basic copyright principles is a two-step process. Step One is to characterize, at the lowest level of abstraction above the work itself, the idea (here, the particular system) associated with a given expression (here, an XML schema or portion thereof). Step Two is to determine whether one could make substantial changes in the expression that would not substantially change the idea (here, again, the system) or compromise the idea's functionality, thus also changing the idea. If no expressive changes can be identified that would leave intact the idea's functionality, or if the proponent of copyright protection acknowledges that there are no such expressive changes, then the idea and expression are, according to such an approach, "merged."[123]

computer programs merely because such programs are functional. *Cf. id.* at 144 (computer programs are protected by copyright). It would seem to follow that the functionality of computer languages is insufficient, by itself, to deprive them of copyright protection, since functional computer languages are at least as "expressive" as functional computer programs.

[123] *Cf.* Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1236 (3d Cir. 1986) ("[*T*]*he purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea.*") (emphasis in original). In *Whelan*, the court concluded that the structure of a particular computer program was not necessary to its function and thus was copyrightable expression. *See id.* at 1239.

Whelan has been criticized for extending copyright protection by focusing on the overall "work's idea" rather than more specific ideas within the work, which may be more likely to merge with particular expressions. *See* Computer Assoc., Int'l v. Altai, Inc., 982 F.2d 693, 705, 44 U.S.P.Q.2d (BNA) 1281 (2d Cir. 1992) ("As we have already noted, a computer program's ultimate function or purpose is the composite result of interacting subroutines. Since each subroutine is itself a program, and thus, may be said to have its own 'idea,' *Whelan*'s general formulation that a program's overall purpose equates with

For example, in Step One, Happy Hour Markup Language could be characterized, at least in part, as a system for marking up Web pages and other documents relating to mixed drinks in a way that specifies elements of the document that contain drink categories, drink names, ingredient types, ingredient quantities, and similar information so that search engines and software agents can locate and process these elements across a wide range of documents that use the system.[124]

In Step Two, one would determine whether one could make a substantial change in the expression of the schema that defines HHML—for example, a change in element names or allowable attributes—that would not substantially change the system or compromise its functionality. Because of the nature of XML schemas and their dependence on XML syntax, substantial changes (*e.g.*, adopting different element names or allowable attributes, and not merely reversing the order of some of the declarations) in the expression almost certainly would compromise the system's functionality.

Indeed, a key part of any XML schema's functionality, as outlined in Step One, is to make it possible to locate and process the defined elements across many different documents using the system, without using a translator or other additional steps. One could change the expression in ways that would result in different systems covering the same subject matter, but they would be just that: *different* systems. This fact indicates that, if the approach outlined here and below is correct, XML schemas ordinarily will "merge" with the systems they embody. The basis for this approach, and possible objections, are addressed in the pages that follow.

the program's idea is descriptively inadequate.").

Here, the issue is not copyright protection for non-literal elements, such as a program's "structure." Nor is it identifying non-literal "expression." Rather, the problem is to determine whether and when literal expression (a schema) merges with an idea. The test outlined above draws on *Whelan's* observation that changing the functionality of a utilitarian work is a change of idea, while at the same time acknowledging, as observed by the *Altai* court, that there may be multiple ideas in a work and multiple levels of abstraction at which an idea can be examined. *See* 982 F.2d at 705, 706.

[124] The system, undoubtedly, could be characterized in greater detail, but still at a level of abstraction above the expression. For present purposes, however, the characterization above should suffice.

A. THE MEANING OF "IDEA"

At the outset, any analysis that delves into the merger doctrine must confront the dichotomy between expression and idea. How to apply this dichotomy is not necessarily obvious.[125] After all, we generally access "ideas" through "expressions" of them. We cannot say what an idea is without in some sense expressing it. We may doubt whether ideas can be said to exist other than in the form of expressions. One reason for these concerns, no doubt, is that "idea" has various meanings. Because differences in these meanings can cause confusion, examining the term may help clarify the issue.

For purposes of the idea-expression dichotomy as set forth in the Copyright Act, "idea" really means "idea, procedure, process, system, method of operation, concept, principle, or discovery. . . ."[126] As discussed above, the most apt of these words for describing an XML schema or other computer language is probably "system." The word "system" is not, of course, a synonym for "idea" in all respects, but "system" falls on the "idea" side of the idea-expression divide.

What places it there, and what "idea, procedure, process, system," and the other specified "idea-side" words all have in common for copyright purposes, is that they are all abstractions from a given expression. As Learned Hand pointed out in another key early copyright decision, *Nichols v. Universal Pictures Corp.*,[127] if we want to say that a given expression, X, represents a particular idea, Y, then Y is an abstraction we construct from X.[128] The abstraction Y is, of course, itself an "expression" of the idea, system, etc., but it is ordinarily different from X and is not the direct expression of the author.[129]

[125] *See* Leslie A. Kurtz, *Speaking to the Ghost: Idea and Expression in Copyright*, 47 U. MIAMI L. REV. 1221, 1222 (1993) ("The words idea and expression remain strangely undefined, terms[ ] without content, bottles without wine."); Edward Samuels, *The Idea-Expression Dichotomy in Copyright Law*, 56 TENN. L. REV. 321, 323 (1989) ("Some commentators who have studied the idea-expression dichotomy in the greatest detail have criticized it. . .").

[126] 17 U.S.C. § 102(b) (1994). *Baker v. Selden* used the word "art," presumably because, in defining patentable subject matter, the Patent Act used the word "art," rather than "process," until 1952. *See* Diamond v. Diehr, 450 U.S. 175, 182 (1981).

[127] 45 F.2d 119 (2d Cir. 1930).

[128] *Id.* at 121.

[129] Whether Y or something like it was also somehow floating around in the mind of the author before he or she expressed X may be an interesting psychological question, but it makes no difference for copyright analysis.

The word "idea" can be traced to the Greek ιδεα.[130] For Plato, an Idea or Form was the essence of a thing (*e.g.*, chairness) as opposed to a particular instance of the thing (*e.g.*, a particular chair).[131] The Form was perfect; the particular instance of the thing was only an imperfect approximation of the Form.[132] Ever since, philosophers have given considerable attention to whether and where ideas "exist."[133] Whereas Plato placed ideas in a world of their own, Aristotle denied that abstract entities had any independent existence.[134] More recently, "concept" (which appears along with "idea" in the Copyright Act's litany of exclusions) has become "the modern replacement for the older term idea, stripped of the latter's imagist associations, and thought of as more intimately bound up with language."[135]

Although "idea" and "concept" have other meanings,[136] a popular dictionary includes a definition of "concept" as "an abstract or generic idea generalized from particular instances."[137] Reflecting this sense of the word, the idea-expression dichotomy in copyright law clearly echoes the classical search for essences, which must be achieved, however, through the

[130] "Idea," in *Philosophy Pages*, *at* http://www.philosophypages.com/dy/i.htm (last visited Aug. 30, 2001); FARHANG ZABEEH, UNIVERSALS: A NEW LOOK AT AN OLD PROBLEM 3 (1966). The suggestion here is not that copyright law depends on philosophical theory, but that the meaning of "idea" as we use it in copyright analysis is significantly influenced by our philosophical tradition, making an ever-so-brief excursion into that tradition potentially helpful in understanding how copyright analysis applies to the issue at hand.

[131] *See* "form," in THE CAMBRIDGE DICTIONARY OF PHILOSOPHY 315 (2d ed. 1995).

[132] *See id.*; *cf.* "idea," in MERRIAM-WEBSTER'S COLLEGIATE DICTIONARY, *supra* note 65 ("1a: a transcendent entity that is a real pattern of which existing things are imperfect representations").

[133] *See* "abstraction," in THE OXFORD COMPANION TO PHILOSOPHY 3 (1995). For example, the existence of the abstract "idea" of a triangle was questioned because, by hypothesis, it encompasses all particular kinds of triangle (equilateral, scalene, etc.) but at the same time is none of these particular kinds of triangle. *Id.*; *see* "abstract entity," in THE CAMBRIDGE DICTIONARY OF PHILOSOPHY, *supra* note 131, at 3.

[134] *See* "abstract entity," in THE CAMBRIDGE DICTIONARY OF PHILOSOPHY, *supra* note 131, at 3.

[135] *See* "concept," in THE OXFORD COMPANION TO PHILOSOPHY, *supra* note 133, at 146.

[136] For example, in the present context, "idea" may also refer to an inchoate or incompletely-formed expression. *See* "idea," in MERRIAM-WEBSTER'S COLLEGIATE DICTIONARY, *supra* note 65 ("3b: an indefinite or unformed conception"). If one thinks before one speaks, one has "in mind" the general nature of what one is about to say, and we might call this mental image the "idea" that one is "trying" to express. In this sense, the idea precedes the expression, perhaps in a causal sense. The author had the "idea" for this Article before writing this Article. The author also had "ideas" for particular points in the Article before expressing them. Such ideas become accessible, however, if at all, only through the expressions in which they are eventually embodied, and they may or may not correspond to the "ideas" that copyright does not protect.

[137] *See* "concept," in MERRIAM-WEBSTER'S COLLEGIATE DICTIONARY, *supra* note 65 (definition 2). This definition roughly corresponds to the definition of "idea" as "the central meaning or chief end of a particular action or situation." *Id.* (definition 6).

construction of abstractions.[138] That is, because copyright doctrine presupposes that idea and expression are *dichotomous*, and because that which is given directly in the expression is part of the expression, "ideas" as we know them for copyright purposes cannot be something that is given directly, but are essences of what is given, which in today's terms we think of as abstractions that we construct. The construction of these abstractions occurs as part of copyright analysis, in which we define or characterize some aspect of the expression being analyzed. The case law confirms that identifying the idea we say is "embodied" in an expression involves this process of constructive abstraction.[139]

Thus, if we are presented with a description of a particular chair, we may look at the chair and state the idea "chairness" even though, looking at the chair, we do not see chairness.[140] Stating the idea involves specifying some features (*e.g.*, capacity to support someone while seated) and ignoring others (*e.g.*, color).[141] Whether abstract entities in any sense "exist" outside the human mind, or even within the mind of a work's author, can safely remain in the realm of metaphysics or perhaps psychology.[142] Likewise, we need not

[138] *Cf.* 1 ALLEN, THE DIALOGUES OF PLATO: EUTHYPHRO, APOLOGY, CRITO, MENO, GORGIAS, MENEXENUS 37-38 (R.H. Allen, trans., 1984) ("Socrates' identification, or connection, of things which are linguistically distinct [as representing ideas] is unintelligible without the notion of essence . . ."); PLATO, THE REPUBLIC 350-51 (Benjamin Jowett trans., 1941) (essence corresponds to truth).

[139] *See, e.g.*, Computer Assoc., Int'l v. Altai, Inc., 982 F.2d 693, 706, 44 U.S.P.Q.2d (BNA) 1281 (2d Cir. 1992); Nichols v. Universal Pictures Corp., 45 F.2d 119, 121, 7 U.S.P.Q. (BNA) 84 (2d Cir. 1930).

[140] *Cf.* D.M. ARMSTRONG, UNIVERSALS: AN OPINIONATED INTRODUCTION 6 (1989) ("Antisthenes said to Plato: 'I can see the horse, Plato, but not horseness.' ").

[141] The end product of this process will itself be an expression of the concept—a kind of meta-expression—because the only way the concept becomes accessible (except perhaps to the person conceiving it in inchoate form in that person's mind) is through expression. Likewise, identifying a procedure, process, system, method of operation, principle, or discovery embodied in an expression involves meta-expression, because we also need to express our identification of the procedure, process, etc. These meta-expressions, however, preserve the dichotomy because they are not directly given by the expression under analysis, but instead are expressions of abstract ideas we associate with the expressions under analysis.

[142] As David R. Koepsell observes, one need not resolve the differences between philosophical realism and idealism to adopt a satisfactory "ontology," a term he uses to describe the appropriate classificatory framework, for examining intellectual property issues relating to software:

> An acceptance of the use of common terms which refer to objects commonly understood, and the relations which may exist between those terms and our experiences, is all we need in order to categorize the world. Developing a common-sense ontology requires an admission that we cannot satisfactorily answer the ultimate metaphysical questions regarding what may be real and what may not be real. Rather, it requires only an acceptance of the facts of common experience, without regard to their deeper reality.

DAVID R. KOEPSELL, THE ONTOLOGY OF CYBERSPACE: LAW, PHILOSOPHY, AND THE FUTURE OF try to solve the various other philosophical and linguistic problems associated with universals.[143] For purposes of applying copyright doctrine, what matters is whether one can find a satisfactory way, consistent with ordinary language and common sense, to construct an abstraction—"idea," "concept," "system," or the like—that provides a satisfactory sense of the relevant "essence" of the expression.[144]

B. LEVELS OF ABSTRACTION

Plato's theory of ideas attracted considerable criticism, even in his own time, and one of the most important criticisms arose from the observation that the number of universals was potentially infinite.[145] The modern version of this problem is that we can associate any number of concepts with a particular thing and any number of ideas with a particular expression. A particular chair may be an instance of the idea "chairness," but it may also be an instance of the idea "furniture." Defining the applicable idea, or the applicable level of abstraction, becomes a key issue.[146]

INTELLECTUAL PROPERTY 24-25 (2000).

[143] Some philosophers believe that Wittgenstein "solved" the problem of universals by exposing it as a philosophical confusion. *See, e.g.*, Benford Bambrough, *Universals and Family Resemblances*, *in* THE PROBLEM OF UNIVERSALS 266 (Andrew B. Schoedinger ed. 1992); ZABEEH, *supra* note 130, at 43. We need not address whether every use of a particular term entails a common "something," however, to recognize that, in a particular given context such as a specific case, subject to analysis under the principles of copyright law, it may make sense to refer to an abstract idea or concept associated with a particular expression. *See* LUDWIG WITTGENSTEIN, PHILOSOPHICAL INVESTIGATIONS § 68 (G.E.M. Anscombe trans., 2d ed. 1958) (even if "number" can be used in various ways, "I *can give* the concept 'number' rigid limits . . . that is, *use* the word 'number' for a rigidly limited concept") (emphasis on "can" in original; other emphasis added).

[144] Referring to the process as one of constructive abstraction does not imply that the abstraction is somehow arbitrary. On the contrary, we must find abstractions that fit the expression. For purposes of copyright analysis, however, we need not adopt a complete theory of what such abstractions really "are."

[145] *See* "Plato," *in* CAMBRIDGE DICTIONARY OF PHILOSOPHY 712 (2d ed. 1995). For the modern view of this criticism, *see* NELSON GOODMAN, THE PROBLEM OF UNIVERSALS (1956):

> [O]ur platonist admits all classes of classes of atoms, and by this single step he welcomes into his universe 2^{31}, or over two billion, additional entities. And he has no thought of stopping there. He also admits all classes of classes of atoms, and so on *ad infinitum*, climbing through an explosively expanding universe towards a prodigiously teeming Platonic Heaven.

Id. at 19, *quoted in* ZABEEH, *supra* note 130, at 24 n.9.

[146] Unfortunately, despite focusing on "ontology" and repeatedly discussing the idea-expression dichotomy, Koepsell states: "I am not interested in pursuing here the nature of ideas. It suffices for the pursuit of a commonsense ontology of intellectual property law to note that an idea may be held and not expressed." KOEPSELL, *supra* note 142, at 121. In other words, an idea, as opposed to an expression, is

In *Nichols v. Universal Pictures Corp.*,[147] the Second Circuit described an approach to the idea-expression dichotomy that has become known as the " 'abstractions' test,"[148] although, as discussed below, it is more of a non-test. The court wrote:

> Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas," to which, apart from their expression, his property is never extended.[149]

This approach is something less than a "test" because the court went on to say, in the next sentence: "Nobody has ever been able to fix that boundary, and nobody ever can."[150] Rather than proposing a test for fixing the "point" in the "series of abstractions where they are no longer protected," the court simply observed that the line needed to be drawn to decide the case and applied an "I-know-it-when-I-see it" approach, concluding: "Whatever may be the difficulties a priori, we have no question on which side of the line this case falls."[151] On this basis, the court held that the two works in question were not substantially similar at a level of abstraction sufficiently specific to be viewed as expression rather than idea.[152]

More than sixty years later, in *Computer Associates International, Inc. v. Altai, Inc.*,[153] the Second Circuit incorporated the *Nichols* "abstractions" approach into its so-called "abstraction-filtration-comparison" test for identifying copyrightable expression, if any, in a computer program's

that which is "not expressed." To the extent this observation is true, it is true by definition; that which is not expressed is not an expression. We still need to know where "idea" ends and "expression" begins.

[147] 45 F.2d 119 (2d Cir. 1930).

[148] Computer Assoc., Int'l v. Altai, Inc., 982 F.2d 693, 706, 55 U.S.P.Q.2d (BNA) 1281 (2d Cir. 1992).

[149] *Nichols*, 45 F.2d at 121.

[150] *Id.*

[151] *Id.* at 122.

[152] *See id.*

[153] *Altai*, 982 F.2d 693.

"structure" or non-literal elements.[154] Under this approach, a program is examined at each level of abstraction. "[A]t "progressively higher levels of abstraction, the functions of higher-level modules conceptually replace the implementations of those modules . . ., until finally, one is left with nothing but the ultimate function of the program."[155]

Next, one follows "a 'successive filtering method' for separating protectable expression from non-protectable material," in which one examines "the structural components at each level of abstraction to determine whether their particular inclusion at that level was 'idea' " or otherwise nonprotectable by copyright.[156] The object is to "sift out all non-protectable material."[157] Finally:

> Left with a kernel, or possibly kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program. The result of this comparison will determine whether the protectable elements of the programs at issue are substantially similar so as to warrant a finding of infringement.[158]

The problem with this strategy for harvesting protectable "kernels" is that, like *Nichols*, it provides no principled basis for separating the wheat of expression from the chaff of ideas. Specifically, the approach gives no basis for determining the level of abstraction or, as *Nichols* puts it, the "point in [the] series of abstractions,"[159] at which "expression" becomes "idea."[160] The approach presupposes that there are relatively "low" levels of abstraction—*i.e.*, relatively concrete statements of the work—that are not sufficiently removed from the expression to be "ideas," so that similarity between two works at those levels would establish infringement despite the absence

[154] *Id.* at 706.

[155] *Id.* at 707 (quoting Steven R. Englund, Note, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 897-98 (1990)).

[156] *Altai*, 982 F.2d at 707.

[157] *Id.* at 706.

[158] *Id.*

[159] 45 F.2d at 121.

[160] *Cf.* Samuels, *supra* note 125, at 343 ("[I]f the abstraction at issue is not determined by the characters or plot as developed, how does one determine the level of an abstraction upon which to focus?").

of literal identity. But such an approach does not show how (other than by knowing it when we see it) we decide the level of abstraction at which to make this dispositive comparison, or above which similarity is in the idea only.

Both *Altai* and *Nichols* suffer from a serious further problem as well. They both view the varying "abstractions" that constitute the characterization of a work's "idea" as merely more generalized ways of expressing the work. Perhaps the word "idea" lends itself to this understanding, but words such as "system" may not. Characterizing the "system" embodied in a work is not necessarily a matter of finding a more general way of expressing the work, but instead entails identifying properties of the work that define its "systematic" functionality.[161] Here, the relevant properties center on the role of the schema as part of a system that itself serves as a means of expression. We need not abstract all possible aspects of the schema to recognize it as such a system and to treat the system as unprotectable. To be sure, an abstracting process is required, but that which is being abstracted is only one aspect of the work—the aspect that allows us to call it a (functional) system.

In *Nichols*, the court suggested that the highest level of abstraction of a play was its title.[162] The title, of course, does not necessarily capture or encompass the full range of more specific ideas in the play. Rather, it usually focuses on a particular idea or aspect of the work selected by the author.[163] When we refer to a "system," the use of this term to describe the kind of concept involved directs us to focus on particular aspects of the work relating to the work's functionality. In this context, it is worth noting that *Altai*'s adoption of the *Nichols* approach in the software context overlooks the *Nichols* court's emphasis that its approach was "especially" suitable for a

[161] That an "idea" or "system" is an abstraction does *not* mean that all abstractions from an expression are excluded from protection. Paraphrasing, for example, can be viewed as an abstraction, as can, perhaps, translation. An abstraction from an expression defines an unprotected aspect of the expression only if the abstraction is a characterization of the embodied "idea, procedure, process, system, method of operation, concept, principle, or discovery." 17 U.S.C. § 102(b) (1994).

[162] 45 F.2d at 121.

[163] For that matter, in rare instances, the title may even contain more of certain kinds of expression than the work itself. For example, in the 1981 Jean-Jacques Annaud film "Quest for Fire," which takes place 80,000 years ago, the cave-people characters speak no known language, but rather use languages that were invented for the film. Although the characters are quite expressive in other ways, the only identifiable modern words in the work are the three that make up the title.

play,[164] which is not a comparably functional work. As seen below, the functional aspect is key to defining a "system."

C. RETURNING TO *BAKER V. SELDEN*

To see how the approach suggested in this Article is derived and how it would address these problems, we may set to one side the 1992 *Altai* decision and the 1930 *Nichols* decision and return to the Supreme Court's 1880 decision in *Baker v. Selden.*[165] The decision in *Baker*, after all, provides the doctrinal foundation for the idea-expression dichotomy that is now codified in the Copyright Act.[166] As we are about to see, it also addresses the key issues in the context of a clearly functional work.

Baker involved a book about an accounting system.[167] The Court acknowledged that the "description of the art" in the book was "entitled to the benefit of copyright. . . ."[168] The book did not, however, consist entirely of the introductory essay that contained this description.[169] Rather, "annexed" to the book were "certain forms or blanks, consisting of ruled lines, and headings, illustrating the system and showing how it is to be used and carried out in practice."[170] The Court held that the specific "system explained in [Selden's] book," namely one keyed to Selden's particular "ruled lines and headings, or ruled lines and headings made and arranged on substantially the same system . . .," was unprotectable by copyright.[171]

The "art" or idea in *Baker* could have been defined, at a relatively high level of abstraction, as an accounting system. At an only partially-lower level of abstraction, it could have been defined as a double-entry accounting system. Selden did not claim copyright at either level of abstraction. Rather, he asserted protection only for his specific system, which Selden himself contended no one else could use "without using substantially the same ruled lines and headings which he has appended to his books in illustration of

[164] *Id.*

[165] 101 U.S. 99 (1880).

[166] *See* Mazer v. Stein, 347 U.S. 201, 217 (1954) (citing *Baker*).

[167] *Baker*, 101 U.S. at 100.

[168] *Id.* at 105.

[169] *See id.*

[170] *Id.* at 100.

[171] *Id.* at 101.

it."[172] In holding, effectively, that Selden's particular expression of the grid merged with the idea, making the grid along with the idea unprotectable, the Court viewed the unprotected idea at what appears to have been the lowest available level of abstraction. In doing so, the Court focused on the work's functionality as a system.

Although the *Baker* Court did not directly address the possibility that it could have defined the unprotected "art" or idea at a level of abstraction that was higher, and thus farther removed from the particular work, the Court emphasized that protection for an art must be found, if at all, in patent law.[173] Given this point, one explanation for the Court's use of the lowest level of abstraction is that allowing protection of a functional idea that can be attributed to a work at *any* level of abstraction would amount to a patent grant,[174] with effective "claims" of undetermined scope. Under this approach, copyright protection for an idea in a functional work that takes the form of a "system" must be denied at the lowest level of abstraction; if denied at that level, it will be denied at any higher level.[175]

The Court's emphasis on the work's functional aspect is seen in its statement that, "in *using* the art [described in the book], the ruled lines and headings of accounts must necessarily be *used* as incident to it."[176] One could substantially vary the expressive aspects of the essay that described the "art" or idea, while still *describing* an idea that, even at the lowest level of abstraction, can be recognized as the same idea. One could not, however, substantially vary the expressive aspects of the ruled lines and headings of accounts without defeating the *functionality* of that idea—they "must necessarily be used as incident to it"—and thus effectively varying the idea.[177] This is the sense in which the ruled lines and headings *are* the unprotected idea.

[172] *Baker*, 101 U.S. 100, at 101.

[173] *Id.* at 102. The Court expressed this view in strong language: "To give to the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters-patent, not of copyright." *Id.*

[174] *See id.* at 102-03.

[175] Drafters of patent claims are quite familiar with this logical relationship. Narrower (more specific) claims are more likely to be granted, but broader (more general) claims cover more ground. The narrower claim may survive denial of the broader claim, but the converse is not true. *See generally In re* Hiniker Co., 150 F.3d 1362, 1369 (Fed. Cir. 1998) ("the name of the game is the claim").

[176] *Baker*, 101 U.S. at 104 (emphasis added).

[177] *See id.*

To return to the classical concept of ideas with which we began the discussion of the idea-expression dichotomy, the word "essence" in contemporary usage still conveys, in the present context, the notion of "essence" as a property of a thing that cannot be changed without causing the thing to lose its identity.[178] If the particular expressive content of a work that embodies a functional system cannot be changed without causing that work to lose its identity as an embodiment of that system, then this particular expressive content *is* the work's "essence," and thus "merges with" the work's relevant "idea."

Just as Selden's "art," in the most concrete sense, was a system keyed to the use of his particular ruled lines and headings or substantially similar ones, an XML schema, as an idea or system, depends on the use of the particular vocabulary set forth in it. One could *describe* the system represented by the schema differently in an essay while recognizably referring to the same system. One could also select, for inclusion in the schema, a different vocabulary to refer to similar concepts, but then one would not be using the same system. Changing the vocabulary, or making almost any other substantial change in the literal expression of the schema, would almost surely defeat the functionality of the system. Rather, one would be creating a new system, which might or might not be functional. In other words, in the same way that Selden's arrangement of lines and headings *was* his system, the XML schema *is* the system it embodies.[179] If this analysis is correct, XML schemas will always, or almost always, be uncopyrightable.

A good test of this approach is to consider whether it explains why the source code for an application program ordinarily does not merge with the idea and therefore is copyrightable. In other words, the approach should indicate how an idea can be associated with more than one expression. For example, just as an XML schema is a language system (or subsystem) that

[178] *See* "essence," in MERRIAM-WEBSTER'S COLLEGIATE DICTIONARY, *supra* note 65 ("1c: the properties or attributes by means of which something can be placed in its proper class or identified as being what it is"); ZABEEH, *supra* note 130, at 12 ("Aristotle argues that . . . the essence of a thing is that property which cannot be changed without the thing losing its identity").

[179] If the work of authorship as a whole (the entire schema) merges with the "system" or other idea that it embodies, there is no need to consider whether pieces of expression within the work merge with the additional ideas they may also embody. This is so because, as discussed above, uncopyrightability of an idea trumps any copyrightability of expressions in which it may be embodied. *See supra* notes 110-11 and accompanying text. On the other hand, as suggested by *Altai*, a work as a whole may embody multiple ideas, and if there is no overall merger, particular expressions within the work may nevertheless merge and thus be uncopyrightable. *See supra* note 123.

enables expression, a text-editing program can be viewed as a "system" (or subsystem) that enables expression. One might say that a particular text-editing program expresses "the idea that only it expresses," in which case (arbitrary) merger may appear to be achieved. We may, however, reasonably exclude such circular or implausible characterizations of the idea.

Rather, for purposes of copyright analysis, we must find a way to characterize the system in functional terms (as the Supreme Court did with Selden's system) that is both linguistically plausible and no more abstract than necessary as a matter of ordinary language. One possibility for a particular text editor would be to say that it is a system that enables the user to create and save files in, say, ASCII format. Clearly, one could substantially change the code (expression) without needing to change this characterization and without impairing the system's functionality (thus changing the system). If there were code that could not be changed without preventing the system from functioning as an editor for ASCII-formatted files, or that is necessary to the functionality achieved by a given subroutine, this particular code probably would merge with the idea. Otherwise, however, the code is only one of many, potential expressions of the idea and is, accordingly, copyrightable.[180]

D. RECENT MERGER DECISIONS

Although *Baker v. Selden* and the principles behind it, which are codified in the current Copyright Act, seem to support the analysis outlined above, current case law has not yet applied a consistent set of principles to such issues, leaving the outcome far from certain. Thus, unless the holdings of recent decisions by federal courts of appeals at some point are rejected by the Supreme Court, the copyrightability of XML schemas under recent decisions may turn on how similar XML schemas are to works such as model laws and

[180] One might object that the approach suggested here means that any software designed to achieve interoperability will be unprotectable. This is not, however, the effect. At most, the approach would mean that the particular portion of a program that allows for interoperability is unprotectable if the interoperability cannot be achieved with other code. For example, in a text editor, if a portion of the code must reproduce certain terms to create a file in a standard format, the portion that consists of the reproduction of those terms will not be copyrightable. But the remaining code, which can be varied without defeating the ability to produce a file in the standard format, remains copyrightable. XML schemas are different because they consist almost entirely of the terms that must be reproduced to create a file in the standard format.

taxonomies of medical procedures, which have been held not to merge with ideas.

1. Model Laws. In *Veeck v. Southern Building Code Congress International, Inc.*,[181] the Fifth Circuit recently considered whether model building codes, once enacted into law, become a "fact" or "idea" that can be expressed in only one way and thus become uncopyrightable.[182] The accused infringer asserted that "there can be only one expression of the law."[183] The court examined the model building codes "at the times of their creation," however, and asked "whether at that instant they merged with the idea of 'building codes' . . . ."[184] The court concluded that they did not, because "there remain many ways to write model building codes, not just one."[185]

The court's opinion leaves unclear precisely what the unprotected idea was alleged to be ("the law," of which assertedly "there can be only one expression," or "model building codes," of which there clearly can be many). Nevertheless, the decision clearly relies on the temporal point, namely that a model code becomes an expression of "the law" only after it is enacted.[186] Likewise, an XML schema may become an industry standard if it gains acceptance, but may not be when created. Is the schema, when created, therefore an expression that has not yet merged with an idea?

There is a key difference between model building codes and XML schemas. A model building code, before enactment, is what one might call "law manqué." It is not capable of "functioning" as law. It may be influential, but does not have the force of law. If people modify their behavior because of it, they presumably do so in anticipation of its becoming

[181] 241 F.3d 398, 57 U.S.P.Q.2d (BNA) 1665 (5th Cir. 2001).

[182] Mason v. Montgomery Data, Inc., 967 F.2d 135, 138 n.5 (5th Cir. 1992). There are differences among the circuits on whether "merger" is a doctrine defining limits on copyrightability or limits on actionable infringement. *Compare Veeck*, 241 F.3d at 407 ("In this circuit, the merger doctrine has been applied to the question whether a work was copyrightable at the time of its creation, preventing a copyright from attaching in the first place, rather than as an infringement defense focusing on merger at the time of copying."), *with* Matthew Bender & Co. v. West Publ'g Co., 158 F.3d 674, 688 n.12 (2d Cir. 1998) (noting that the Second Circuit treats the merger doctrine as relating to whether there has been actionable infringement), *cert. denied*, 522 U.S. 3732 (1999).

[183] *Veeck*, 241 F.3d at 407.

[184] *Id.*

[185] *Id. See also* CCC Info. Servs., Inc. v. MacLean Hunter Mkt. Reports, Inc., 44 F.3d 61, 74, 33 U.S.P.Q.2d (BNA) 1183 (2d Cir. 1994) (stating that a state's reference to copyrighted work in legal standard does not deprive work of copyright protection); *but cf.* Banks v. Manchester, 128 U.S. 244, 253 (1888) (stating that judicial opinions are not copyrightable).

[186] It may later become "the law," but *Veeck* indicates that a work, once protected, cannot later be "denuded" of copyright protection under the merger doctrine. 241 F.3d at 407.

law, but such action is only a matter of anticipation. Because the model code is not law when it is fixed, copyright protection attaches and (according to the decision) cannot later disappear.

In contrast, a well-formed XML schema, from the instant it is created, can "function" as part of a markup language.[187] If the markup language is an unprotectable "system," merger has occurred at the time the expression of the schema is fixed. The schema can be used as part of or together with an XML document and an XML parser, and the schema immediately will work, if it is well-formed. There is no need for any contingent event, such as legislative enactment. Industry standardization may determine how widely the schema is used, but is unnecessary to merge schema with system.

2. *Industry Classifications.* In *Practice Management Information Corp. v. American Medical Association*,[188] the accused infringer maintained that a set of medical procedure codes that had been mandated by the federal government for use in Medicaid applications became "an uncopyrightable industry standard or 'idea.' "[189] The Ninth Circuit disagreed, stating: "[The AMA's copyright in the procedure codes] does not prevent Practice Management or the AMA's competitors from developing comparative or better coding systems and lobbying the federal government and private actors to adopt them. It simply prevents wholesale copying of an existing system."[190]

Although the Ninth Circuit rejected AMA's infringement claim based on copyright misuse, the court's language on the merger doctrine appears inconsistent with *Baker* and with Section 102(b) of the Copyright Act. The Court in *Baker* would have reached the opposite result had it observed, a là the Ninth Circuit in *Practice Management*, that Selden's copyright did not prevent Baker, or Selden's competitors, from (in the Ninth Circuit's words) "developing . . . better . . . systems." Instead, the Court made clear that Baker and others were entitled to use Selden's specific system, so long as Selden lacked a patent. Likewise, Section 102(b) specifically bars copyright

[187] If the schema does not correspond to the applicable requirements of XML, it may not function until it is corrected. It seems counterintuitive, however, that defects in a functional work would increase its copyrightability. The doctrinal answer is that the "idea," or here the "system," is an abstraction from what the work purports to be, not a critical commentary on how well the work fulfills its purpose.

[188] 121 F.3d 516, 43 U.S.P.Q2d (BNA) 1611 (9th Cir. 1997), 522 U.S. 923 (1997), *modified on other grounds*, 133 F.3d 1140 (9th Cir. 1998).

[189] *Id.* at 520 n.8.

[190] *Id.*

protection for "systems," making it unclear why the Ninth Circuit defended copyright to prevent copying of a "*system.*"[191]

Despite these infirmities in the Ninth Circuit's analysis of the merger doctrine in *Practice Management*, the Seventh Circuit adopted a similar approach in *American Dental Association v. Delta Dental Plans Ass'n.*[192] There, the Seventh Circuit considered whether a "taxonomy" of dental procedures was copyrightable.[193] Addressing an argument that the expression in the taxonomy merged with facts or ideas, the court asserted that "Einstein could have explained relativity in any of a hundred different ways; another physicist could expound the same principles differently."[194] The court continued:

> So too with a taxonomy—of butterflies, legal citations, or dental procedures. Facts do not supply their own principles of organization. Classification is a creative endeavor. Butterflies may be grouped by their color, or the shape of their wings, or their feeding or breeding habits, or their habitats, or the attributes of their caterpillars, or the sequence of their DNA; each scheme of classification could be expressed in multiple ways. Dental procedures could be classified by complexity, or by the tools necessary to perform them, or by the parts of the mouth involved, or by the anesthesia employed, or in any of a dozen different ways. The Code's descriptions don't 'merge with the facts'

[191] *Id.* The Ninth Circuit's unselfconscious use of the word "system" in the quoted passage to describe the subject matter of the bar on "wholesale copying," and its treatment of the merger issue in a footnote, suggests that Section 102(b) may not have been the court's primary concern, perhaps because the court ruled against the AMA on the basis of copyright misuse. *See id.* at 520. The court distinguished *Sega Enters. v. Accolade*, stating that, in *Sega*, the court had "not allowed the owners of copyrights in expressions mandated by industry standards to use their copyrights to stifle independent creative expression in the industry." *Id.* at 520 n.8 (citing *Sega*, 977 F.2d at 1523-24). *Sega*, however, was a fair use case, not a case that dealt with copyrightability. *See* 977 F.2d at 1514. Although there undoubtedly will be important fair use issues in relation to XML schemas, such issues only arise if XML schemas are copyrightable in the first place. This Article focuses on the copyrightability question, leaving fair use to another day.

[192] 126 F.3d 977, 44 U.S.P.Q.2d (BNA) 1296 (7th Cir. 1997).

[193] *Id.* at 977.

[194] *Id.* at 979.

> any more than a scientific description of butterfly attributes is part of a butterfly.[195]

Although Einstein could have used various different words to describe relativity, the Seventh Circuit in *Delta Dental* acknowledged that "many of the core equations, such as the famous E=mc2, express 'facts' and therefore are not copyrightable."[196] This distinction is both intriguing on its own terms and potentially important for determining whether the merger doctrine applies to XML schemas.

On its own terms, the distinction suggests that an equation expressing a fact is not copyrightable, but a verbal explanation of the same fact is copyrightable, presumably because of the greater pliability of ordinary non-mathematical language. There may, however, be more than one way to express an equation or more than one equation that describes a "fact." Moreover, Section 102(b) of the Copyright Act explicitly states that an idea is unprotectable "regardless of the *form* in which it is described." *Delta Dental* seems to say that the *form—i.e.*, mathematical rather than verbal—makes the difference.

Accepting *Delta Dental*'s holding at face value, does it support the view that an XML schema is copyrightable? In *Delta Dental*, the taxonomy in question was a description of dental procedures. Notwithstanding the court's colorful expression, the basic idea seems to be, essentially, that there are plenty of (accurate) ways to describe a given dental procedure, in contrast to phenomena for which only one equation will do. This idea seems consistent with the approach outlined above, in the sense that there are multiple ways to write source code for a particular ASCII text-editing system, but multiple vocabularies would produce different XML schemas.

Nevertheless, *Delta Dental* may appear difficult to square with the proposition, advanced in this Article, that merger occurs whenever substantial changes in a "functional" expression (like an XML schema) would substantially compromise the functionality of the system it expresses (thus changing the idea). The dental taxonomy, in addition to describing procedures, provided a coding system. As the court said: "No one would read the ADA's Code for pleasure; it was designed and is used for business (for records of patients' dental history or making insurance claims) rather

[195] *Id.*

[196] *See id.* at 979.

than aesthetic purposes."[197] If one were to change the code numbers, or to draw the lines between procedures differently, one would have a similar system, but, arguably, it would not be the same system.

On the other hand, although the medical and dental classifications in *Practice Management* and *Delta Dental* presumably could be and very likely are used in connection with computer applications, the courts in those cases did not analyze either set of classifications as "systems" in the form of computer languages or language extensions. Nor did the courts devote any significant explicit attention to whether the particular expression chosen was essential to the functioning of such systems, other than to point out that competitors remained free to develop alternative classification schemes.

VI. CONCLUSION

The copyrightability of XML-based markup languages is a special case of the larger issue of computer language copyrightability. In turn, language copyrightability requires us to descend into the idea-expression conundrum and helps test whether the very construct of an idea-expression dichotomy, as essential as it is to copyright doctrine, can be applied in a manner that anyone might find right. This test occurs in a context charged with practical economic importance. Despite the economic consequences, however, the merger doctrine is so ingrained in copyright law that it has remained largely unaffected by the trend of the last century toward increasingly detailed copyright legislation. Courts must work out the detailed effect of the broad principles it reflects.

Computer languages are X% actual expression (the specification or schema) and Y% potential expression (the sentences that can be written in the language), where Y is orders of magnitude greater than X. One who advocates copyright protection for XML schemas in particular or computer languages in general must offer a conception of copyright in which protection effectively reaches beyond the narrow X to embrace the vast Y. In doing so, the advocate for such protection must also reconcile this result with copyright values. "It is," after all, "precisely [the] growth in creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was

[197] *Id.* at 978.

intended to promote."[198] Such dissemination not only provides an avenue for an author's expression but also gives rise to the freedom and practical ability necessary for the public to receive the author's ideas.

In the case of XML schemas, the copyright advocate can point out that the schema itself is an original work of authorship, of which a schema user must make an exact or substantially similar copy to make use of the schema. Thus, before the schema user can use X to create Y, the user must be free of copyright impediments to reproducing the essential X. If this result is unsatisfactory, the copyright advocate may say, then the user can always create another schema, Z.

Under the merger doctrine, however, the opponent of copyright protection for schemas can reply that, because systems and other ideas are uncopyrightable, there should be no need to create Z, which in any event is not the functional equivalent of having access to X. Instead, according to the opponent of schema copyright, the schema user is free to use X based simply on showing that X stands equally for both a particular expression and a particular idea—in this case, an idea that consists of a system, the functionality of which depends on keeping its expression exactly the way it is. On this view, intellectual property protection for XML schemas, if any, is available through patent law, subject to its rigorous requirements, and does not automatically arise as a matter of copyright.

This Article suggests that, although the question undoubtedly warrants broad and continuing consideration, XML schemas exemplify the manner in which computer languages force the issue of idea-expression merger. In relevant respects, such languages are more like the equations of physics, at least as such equations were viewed by the Seventh Circuit in *Delta Dental*, than they are like verbal explanations of physical phenomena or verbal classifications of medical and dental procedures. The key difference lies not that in their more "technical" nature, but in the functionality of a particular computer language as a "system" and the dependence on that functionality of the particular expression of the language's vocabulary and syntax.

This functionality and its dependence on a particular expression uniquely defines a language, from the moment its vocabulary and syntax are fixed, in a way that most verbal descriptions or narratives do not. If this analysis is correct, XML schemas are a classic example of merger, with the result that

[198] *Sega*, 977 F.2d at 1523.

both the X and Y of the expanding universe of potential XML expressions are free of copyright constraints on using the schemas that are needed to make such expressions meaningful. If XML is blank Esperanto, XML schemas, which fill in the blanks, are the key to this meaning. The blanks in copyright doctrine, on the other hand, are harder to fill in, and as Learned Hand also foresaw, the debate inevitably will go on.