"Electronic Books and the Open eBook Publication Structure"; Allen H. Renear and Dorothea Salo.
Chapter 11 in *The Columbia Guide to Digital Publishing*, William Kasdorf (ed)
Columbia University Press, 2003. **Final MS**.

# 11

# Electronic Books and the Open eBook Publication Structure

**Allen Renear**

University of Illinois, Urbana-Champaign

**Dorothea Salo**

University of Wisconsin, Madison

Electronic books, or *e-books,* will soon be a major part of electronic publishing. This chapter introduces the notion of electronic books, reviewing their history, the advantages they promise, and the difficulties in predicting the pace and nature of e-book development and adoption. It then analyzes some of the critical problems facing both individual publishers and the industry as a whole, drawing on our current understanding of fundamental principles and best practice in information processing and publishing. In the context of this analysis the Open eBook Forum Publication Structure, a widely used XML-based content format, is presented as a foundation for high-performance electronic publishing.

**[Note: The Open eBook Forum (OEBF) later became the International Digital Publishing Forum (IDPF) and the Open eBook Forum Publication Structure (OEBPS) eventually evolved into EPUB. See idpg.com and the Wikipedia entry for EPUB..
This note added August 2012. – ahr]**

## Introduction

Reading is becoming increasingly electronic, and electronic books, or e-books, are a big part of this trend. The recent skepticism of trade and popular journalists notwithstanding, the evidence shows that electronic books are making steady inroads with the reading public, that the technological and commercial obstacles to adoption are rapidly diminishing, and that the much-touted advantages of electronic books are in fact every bit as significant as claimed. While opinions certainly vary widely on the exact schedule, most publishers believe that it is only a matter of time before e-books are a commercially important part of the publishing industry.

Nonetheless, becoming involved in electronic book publishing can be a daunting prospect for publishers. For one thing, current publishing technology and workflows, even if already computer-based, are typically not engineered to support electronic publishing in general, let alone e-book publishing in

particular—in fact, these workflows are rarely based on sound principles of electronic publishing and document engineering. In particular, the prospect of trying to deliver content from a single data store not only through a variety of different delivery modalities (paper books, e-books, Internet Web sites, databases), but also into a variety of different electronic book reading systems (specific combinations of commercial hardware and software, often with distinctive data formats), looks particularly difficult and costly to publishers. Moreover, however rapidly they may be growing, e-book sales are still a very small portion of publishers' total revenue, and therefore investments in major process redesign are unlikely to be recovered in the next one to three years—all of which adds more uncertainty to deciding when, and how, to begin electronic book publishing.

Yet for many publishers, now may actually be the best time to begin to develop digital-product-oriented workflows and formats:

1. before further temporary fixes to obsolete production processes are made,
2. before even more legacy data (which will require expensive conversion) is created,
3. before more in-house staff are trained in obsolete techniques,
4. before more inappropriate partnerships are made, and unnecessary contracts are executed (with conversion providers, composition houses, and distributors, etc.),
5. while the cost of analysis and systems (re-)design is still fairly low, and
6. before any strategically important competition in the e-book arena has really begun.

As described in the chapter on markup, financially sustainable high-performance electronic publishing can only take place within a framework of standards for content, structure, and presentation; a framework that itself must be based on sound fundamental principles of information processing. Without such a foundation a variety of punishing limitations will prevent the development of the interoperability, functionality, and efficiency required for the commercial success of electronic products.

This is not only an issue for the success of individual publishers, but also for the general prospects for a thriving industry. Recognizing this, an organization of publishing industry participants has created the *Open eBook Publication Structure* (OEBPS), an XML-based content specification for electronic book content. This specification is now fairly widely used throughout the publishing industry as an interchange format for handheld electronic books, particularly in mass market and trade publishing. There is also a real possibility that the OEBPS could eventually become a standard integrating framework for a wider range of electronic publishing, including markets such as textbook and STM (scientific, technical, and medical), and even non-book-like delivery mechanisms such as the World Wide Web.

This chapter first introduces the notion of electronic books, reviewing its history and discussing some of the conceptual problems involved in thinking about what is or isn't an electronic book. It then reviews the advantages that e-books offer and discusses the difficulties in predicting the pace or nature of e-book development and adoption. This introduction concludes with an account of why, despite recent problems in getting the industry started, and despite the difficulties in predicting the near future, we can nevertheless be confident that electronic books will soon be a big part of our reading lives.

Next we analyze some of the specific critical problems facing both individual publishers and the industry as whole, problems that account for the difficulty in creating a flourishing e-book industry that delivers on the original promise of electronic books. The OEB Publication Structure is then presented as a carefully designed solution to these problems, one that not only reflects fundamental principles and best practice in information processing and publishing, but that also judiciously reconciles competing demands in a way that reflects the practical and business realities of electronic publishing. The chapter closes with a brief review of current e-book products and a guide to further reading.

First though, so that the forest does not get lost in the trees, we begin with a preliminary summary of the key features of the OEBPS.

# OEBPS in a nutshell

This overview is designed to be useful to anyone with a general knowledge of electronic publishing issues and can be used, "standalone," to rapidly convey the fundamental nature of this specification. Everything mentioned here is discussed in greater detail (with references, acronym expansion, and descriptions of component standards) later in the chapter.

Note that in what follows "OEBPS" is for the most part used to refer to the OEBPS version 1.2, although we also make reference to some features of 2.0, now under development.

## Basic description of OEBPS

### What it is

OEBPS, the Open eBook Publication Structure, is "an XML-based specification for the content, structure, and presentation of handheld electronic books"; conformance to the OEBPS specification is defined both for e-book *content* (OEBPS "Publications" and "Documents") and for e-book *processors* (OEBPS "Reading Systems").

### What it does

OEBPS enables publishers to create content in a single format that can then be rendered on a variety of reading devices.

### Who developed it

OEBPS was developed, and continues to be maintained and improved, by a large inclusive group of publishing industry participants: publishers, software developers, hardware manufacturers, distributors, services providers, trade associations, public interest groups, and others. This group is now organized as the Open eBook Forum and membership is open to all interested organizations.

### What it is for

The principal objective of the OEBPS is to contribute to the creation of a commercially and socially valuable electronic publishing industry by providing a common format for digital content.

From the perspective of the industry as a whole, having such a single common format will improve the interoperability and functionality of both content and reading software (increasing consumer confidence and satisfaction), lower overall development and processing costs, and support innovation and competitive differentiation—all things that are needed to develop a thriving industry.

From the perspective of the individual publisher there are similar immediate benefits in the interoperability and functionality of their products, opportunities for more efficient relationships with partners, and improvement in efficiency of production processes, particularly in the areas of integration with existing in-house workflows and for supporting multiple products and multiple delivery mechanisms. In fact, some publishers today are beginning to use OEBPS even though they have no immediate plans to publish e-books, because it is an effective way to organize and exchange digital content in general.

### Why we need yet another specification

The practical development and use of electronic publishing content requires not just specialized individual standards (e.g., XML, CSS, Dublin Core), each based on sound principles of information

processing, but it also requires an agreement on a coordinated selection, combination, and application of all these various relevant individual standards. OEBPS provides this framework: comprehensive, constrained, standards-based, and reflecting fundamental principles of information processing and publishing best practice.

### What it is, in a little more detail

OEBPS specifies a coordinated application of established electronic publishing standards, most importantly XML, CSS, XHTML, namespaces, MIME types, Dublin Core, and Unicode. In addition it also defines a packaging method (expressed in an XML Document Type Definition, or DTD) to ensure a consistent and complete organization of content and metadata, and a system to support alternative renderings when nonsupported types of data are included. Under development for OEBPS 2.0 are an XPointer-based linking mechanism, an XPointer-based navigation system, modular support for arbitrary external metadata (in 2.0), and a number of further constraints necessary to ensure interoperability across devices with varying capabilities. In the selection and combination of these standards OEBPS was specifically designed to support innovation, functionality, and competitive differentiation, without sacrificing interoperability.

## Some very important facts about OEBPS

The following are aspects of OEBPS that are critical to understanding its role in the publishing production process and the current direction of its development.

### Specialized XML vocabularies are encouraged

OEBPS is specifically designed not only to allow, but to encourage the use of specialized XML encoding vocabularies (tag sets). This enables consistency and continuity with in-house XML encoding, or with standardized domain-specific XML vocabularies (such as TEI, DocBook, or ISO 12083), as well as allowing innovation and advanced functionality beyond what can be achieved with Extensible HyperText Markup Language (XHTML) markup—or any other prespecified general encoding vocabulary. To ensure that allowing specialized XML vocabularies doesn't reduce interoperability, OEBPS has two requirements: that style rules are specified for all XML elements other than those from the OEBPS Basic Document Vocabulary subset (see next item); and if alternative style rules are specified, at least one is from the OEBPS style language (basically a large subset of CSS2, with some additional constructs). Reading Systems must use these style rules when rendering specialized XML markup and all Reading Systems are required to process the OEBPS style language. These restrictions ensure that all OEBPS content can be presented by all OEBPS Reading Systems even if that content contains specialized XML vocabulary.

### XHTML gets special consideration

In addition to allowing the use of any XML vocabulary (as long as style rules from the OEBPS Cascading Style Sheet, or CSS, subset are provided), OEBPS identifies a large subset of XHTML 1.0 elements and attributes that may be used without style rules; OEBPS processors are required to render these, in the absence of style rules, according to the formatting semantics described in the XHTML 1.1 specification. This ensures that publishers who wish to can quickly and easily exploit existing HTML-based content, tools, and expertise. In addition, and most important, it allows the use of HTML elements—such as those for tables—that may be difficult to format correctly using CSS style rules alone.

## OEBPS is used in several different ways

OEBPS is well suited for any of three different roles in the publishing process: a revisable archival format; an interchange format; and a native processing format. Probably the most popular commercial use of OEBPS at this time is as an interchange format: publishers convert their in-house formats into OEBPS, and then in turn convert OEBPS into the various proprietary e-book formats (e.g., Microsoft Reader's .lit, RCA eBook's .rb, or Palm's .pb). This reduces (from $m \times n$ to $m + n$) the number of conversions necessary to provide content that exists in various development formats ($m$) to all available device formats ($n$). It also allows a single format that may be exchanged among content developers, publishers, composition houses, conversion specialists, content licensees, or other partners. Although use of OEBPS as an interchange format is the most common use, some publishers use OEBPS as their standard in-house (revisable) archival format, and, in addition, a number of Reading Devices process OEBPS directly.

## Processing conformance targets *reading systems* not *reading devices*

As described above OEBPS content may be converted into another format before presentation to the human reader. To accommodate the full variety of such configurations the OEBPS, when it specifies processing conformance, takes as its subject the Reading System and not the Reading Device. OEBPS defines the Reading System as, "A combination of hardware and/or software that accepts OEBPS Publications and makes them available to readers." It defines the Reading Device as, "The physical platform (hardware and software) on which publications are rendered." The Reading System includes the Reading Device, and it may be identical with the Reading Device, but it may also extend beyond the Reading Device to include "processing and transformation of data at times prior to, or in locations distant from, its presentation to a human reader."

## OEBPS does not *necessarily* compete with Adobe's PDF

Currently in the publishing industry there is a rough division between e-book strategies that are based on OEBPS and those based on Adobe PDF. However, the preceding points should make it clear that this division, although an empirical fact at present, is not a necessary one. Just as OEBPS content is currently often converted to various proprietary binary formats before presentation on a handheld Reading Device, it might similarly be converted to Adobe PDF for presentation. In fact most general SGML/XML-based publishing (for instance, in STM journal publishing) does routinely produce Postscript or PDF from SGML/XML content as a natural product of the typesetting, or composition process; so it is unlikely that there are any decisive technological or practical obstacles to a similar strategy in e-book publishing. (Adobe Systems is an active member of the Open eBook Forum, an OeBF "Gold Sponsor," and holds a seat on the OeBF Board of Directors.)

## Multimedia and active content is allowed, with conditions

Multimedia and "active" content, because of their variety, complexity, and processing demands, pose the greatest challenge to OEBPS's efforts to reconcile functionality and interoperability in the context of current hardware, software, and data formats. At present (in OEBPS version 1.2), a compromise mechanism of fallbacks is defined to allow the inclusion of multimedia and active content without undermining interoperability. It works like this:

1. Conformant OEBPS Publications are allowed to include multimedia/active content of any kind, but OEBPS Publications that do include such content must also specify a "fallback" to alternative content in a supported format (such as XML text, or a JPEG or PNG image).
2. OEBPS conformant Reading Systems *may* process and render multimedia/active content, but they are not *required* to—they are, however, required to process and present the alternative fallback content whenever they cannot process encountered multimedia/active content.

This strategy is a compromise. It does allow high-performance multimedia and active content e-books, and it does ensure a base level of interoperability for them; but it does not ensure the optimal level of interoperability because it does not guarantee that multimedia/active content will be rendered *as such* by *every* OEBPS reading system. Future versions of OEBPS will have improvements in this area.

## OEBPS Publications can be easily created from XML documents

Producing OEBPS Publications from existing XML content can sometimes be an extremely simple process. For example, suppose that the content for an electronic book exists in several XML text documents that use an in-house XML vocabulary, and that there is a CSS stylesheet that specifies the correct formatting for this content and which does not use any of the CSS constructs that were excluded from the OEBPS subset. An OEBPS Publication would in this case simply consist of these XML documents, the file containing the CSS stylesheet, and an OEBPS Package file; although OEBPS does place several further constraints on XML documents, and doesn't support all of CSS2, it is still quite possible that no, or few, changes to either the XML files or the CSS file will be necessary to "convert" these files to OEBPS. A Package file must be supplied, but this file (itself an XML document defined by the OEBPS Package DTD) can be created simply by adding the following information to an existing blank Package template:

1. A unique identifier.
2. The author of the Publication.
3. The language of the Publication.
4. A list of the files that constitute the Publication, with their MIME media types (in this case that list would be the several XML content documents and the CSS stylesheet; the media types would be "text/x-oeb1-document" for the documents and "text/x-oeb1-css" for the CSS stylesheet).
5. A default order for presenting the XML documents.

Additional information is supported as optional, of course, but this is all that is typically required for a Publication that does not make use of unsupported media types or unsupported style rules. More complicated publications, such as those making use of style rules or media types not supported by OEBPS, are not significantly more complicated to organize as OEBPS Publications. For example, if a file consisting of content in an unsupported media type (e.g., a video clip) is referenced by a Publication, then a fallback file, containing an alternative version of that content in a supported media type (e.g., a JPEG image) must be also included; both files will be listed in the Package file, and the entry for the unsupported media type will indicate which file is to be used for its fallback. Similarly, if a stylesheet using unsupported advanced features of CSS is used, it must be correctly identified and a fallback stylesheet using only supported CSS must be included and correctly identified. Of course it may not always be easy to identify satisfying fallback content in supported media types, or to write a fallback CSS style rule to back up unsupported constructs, but the OEBPS format itself does not add any particular complexity to this process. In short, the production of OEBPS Publications is often a simple matter for publishers already creating SGML or XML content.

## OEBPS 1.2 substantially improves formatting capability

OEBPS 1.2 responds to the highest priority request received from publishers after the first two years of experience with OEBPS versions 1.0 and 1.0.1: increased control over presentation. Toward this end OEBPS 1.2 provides a very substantially augmented set of CSS2 properties, values, and selectors. In addition it enlarges the OEBPS subset of XHTML elements and attributes (while removing deprecated elements and attributes), and improves namespace support.

### OEBPS 1.2 is widely used in the publishing Industry

OEBPS 1.2 is already very widely used throughout the publishing industry and may be considered the "industry standard." This dominance is supported by the existence of conversion tools both for transforming various content development formats into OEBPS and for converting OEBPS into particular e-book device formats. Of course, not all e-book content production is based on OEBPS; some workflows move content directly from non-XML development formats into device formats.

### OEBPS 2.0 will bring further improvements

OEBPS 2.0 will provide major improvements in several areas:

*Metadata:* a mechanism for allowing OEBPS Publications to include metadata resources external to the package file;

*Internationalization:* additional support for providing information about required glyph sets and writing systems;

*Inter- and intra- document linking:* more powerful XPointer-based hypertext linking;

*Navigation:* an extensive XPointer-based navigation mechanism external to the package file, allowing multilingual, graphical, and audio support as well as general high-function navigation (this navigation mechanism is already being used in the DAISY Consortium's Talking Book standard, NISO Z39.86);

*Accessibility:* the foregoing improvements (and those regarding navigation particularly) are wherever possible implemented in ways that improve the general accessibility of e-books to readers with perceptual disabilities.

# Electronic books in general

## E-books: Brief history and future of the idea

Written language is of course fundamental to society as we know it, and imagined devices for enhanced reading have a history that long predates modern electronic computing. More modern references in imaginative literature and in twentieth-century science fiction further attest that technologically enhanced reading was a broadly compelling idea long before, and well apart from, its current appearance in connection with electronic digital computing.

The use of computers for reading is the central theme of the work of the three titans of modern computer-based communication: Vannevar Bush, Douglas Engelbart, and Ted Nelson—all of whom imagined and designed (and in the case of Engelbart and Nelson, actually built) influential systems for electronic reading. Later, the prescient *Network Nation* (Starr and Hiltz, 1978) also had electronic reading as the key benefit of "computer-based communication," describing many of the practices and features we now take for granted and providing a still-relevant analytic framework for comparing the advantages of oral, paper, and electronic interactions.

The earliest working systems for computer-based reading, designed in the 1960s and 1970s (well in advance of both microcomputers and large scale computer-to-computer networking), were typically developed and optimized on (local) networked engineering workstations. Then, because networked workstations were so rare and expensive, such systems were deployed in production versions on mainframes accessible by remote terminals. Seminal work along these lines continued throughout the 1980s, developing both further seminal prototypes such as Notecards, Intermedia, and Microcosm. Eventually, in the mid-1980s to 1990s, commercial hypertext systems that promoted many of the features

and technologies in hypertext and multimedia systems that we take for granted today were released; these included OWL's Guide, Apple's Hypercard, and EBT's DynaText.

Also in the late 1970s and 1980s, research attention turned to the possibility of distinctively *book-like* electronic reading on portable devices [Kaye and Goldberg 1977, Yankelovich et al. 1985].

In the early 1990s the revolution—the World Wide Web—occurred. Computer-based reading with hypertext links and access to vast amounts of material suddenly became widespread and compelling. Unfortunately, almost all of the research achievements of the preceding twenty years, amounting to a tremendous knowledge of best practice and fundamental principles, were ignored, and the Web was, and still is to some extent, a study in bad design. These failures ranged from linking functionality—one direction, untyped, point-to-point, embedded linking—to the underlying markup, which ignored the separation of presentation and content.

In the 1990s electronic books also moved out of the lab and into the marketplace, as actual hardware, reading systems, and content all began to proliferate, providing researchers with actual experience to reflect on and incorporate into their development efforts. Here again, it seemed as if the lessons of the 1960s to 1980s were more often ignored than exploited: early e-books were disappointing in their functionality. Of course, basic hardware and software limitations were equally to blame.

Today (2002) we take reading on-screen for granted, and it may be conjectured that in the developed world most reading (measured simply by the number of words read from one source or other) is probably already electronic. Book-like electronic reading in particular is a rapidly growing commercial phenomenon, with a wide variety of devices, software, and distribution systems, and a wide range of content genres.

## Terminology and scope: What is an e-book?

There are some obvious terminological issues here, and some conceptual ones as well.

First, the very notion of an "electronic book." Do we mean the hardware, the software, the digital content, or some particular combined selection from those three things? For the moment, because the ambiguity of the phrase "electronic book" (and the corresponding neologism e-book) is actually useful and usually clear enough in context, we won't at this point settle on a particular sense. Later, when more clarity is needed, and after we have developed the conceptual framework needed to provide it, we will present terminology that attempts to more precisely match up words and concepts.

Apart from terminological uncertainty, one might also wonder just how "book-like" the content, the software, or the hardware must be to count as an electronic book. We seem to have a rough common understanding of what a characteristic scenario of e-book reading would be: say a high school student reading *Moby Dick* on a handheld device such as a PDA. The paradigm case is thus what reading researchers call "immersive" reading: the reader reads predominantly textual material in a predominantly linear manner (left to right, top to bottom), frequently losing oneself in the flow of the narrative.

This notion of e-book reading, an important one in general, has dominated much recent commercial thinking due largely to the attractiveness of business opportunities in mass market publishing (romance novels, "techno-thrillers," science-fiction and fantasies, horror, mysteries, popular novels, and the like). Here where reading seems to be largely immersive, the technological demands are modest (no hypertext or multimedia needed), and the revenue potential is probably rather greater than with reading for instruction or information.

But the boundaries of the e-book concept remain fuzzy, and the significance of this is more than academic.

Obviously e-book content need not be only fiction or other "aesthetic" reading. A textbook on history, or even physics, could be an e-book. And, almost as certainly, so could a mail-order catalog.

Can e-books contain multimedia or interactive content? Of course they can—and yet we wouldn't describe the appreciation of pure audio and video content (movies and music) lacking even auxiliary verbal content as "reading an e-book," even if the content were presented on a notebook computer, book size tablet, or PDA. What about a movie with just a little additional textual material, say links to persons and events, timelines, and so on.

What about art catalogs and financial data e-books? Can a desktop computer be (or be used as, or be used to read) an e-book? Can e-book content be served in real time over a network? How one draws these boundaries will have consequences for how we think about publishing products and strategies, but there is no advantage in ruling out possibilities by definition, so, again, we will leave these questions open until more precision is needed.

## Advantages of electronic books

Electronic books can provide a number of advantages over paper books.

### Capacity

A person cannot comfortably carry more than a dozen or so paper books, but hundreds or even thousands of books may be carried on digital e-book devices and on lightweight removable storage media.

### Manufacturing

The mass production of multiple instances of paper books is an expensive and complicated process, but electronic books may be produced quickly and inexpensively, simply by copying digital files onto new physical media. And for this reason also electronic books need never be out of print.

### Distribution

Distributing paper books requires physical transportation, warehousing, and shelving. Electronic books can be quickly and inexpensively distributed over the Internet. This is particularly important to regions that are distant from book manufacturing locations, or that don't have the local infrastructure for supporting face-to-face consumer transactions or for reliable and efficient transportation and warehousing.

### Cost

The above advantages suggest that the total cost of e-books, to consumers, might become less than that of paper books. (On the other hand, many factors are in play in determining consumer pricing of e-books, ranging from the very substantial remaining costs such as acquisition and editing that are unaffected by digital electronic production, to the complex business dynamics in the publishing industry that resist any pricing changes that might result in lower net profit.)

### Intelligent viewing

Computer-based presentation of text supports intelligent and customizable viewing. Outlining with expandable entries is the obvious and familiar example, but there are many other possibilities as well. For instance, advanced sections of a textbook, or sections of a manual that don't pertain to the product currently being repaired, can be hidden (perhaps marked with a clickable icon that expands the section in place), as can, in a play script, all parts except for those belonging to the actor holding the book. Colors, indentation, and font shifts can also be used selectively to systematically highlight features of current

interest. Readers (as well as teachers, managers, or others) may specify these changes to suit their particular needs.

### Intelligent navigation

Navigation can be similarly enhanced, allowing the reader to advance through a text in terms of natural logical units (sections, paragraphs, sentences, equations, problems, procedures, etc.) instead of arbitrary media-specific ones such as pages and columns.

### Hypertext

Even though hypertext might be considered be a kind of navigation, or viewing, it is sufficiently powerful and compelling to deserve individual mention. All users of the Web are today familiar with following links from mentioned places, persons, bibliographic references, and the like. And within STM publishing the commonly used linking service CrossRef has had a tremendous impact on how scientists use technical journals. The current state of hypertext functionality on the Web, and in most reading systems, remains very crude, but it is already extremely valuable to readers, and will in any case be undergoing rapid improvement over the next few years as XLink and XPointer technologies become widespread.

### Retrieval

All contemporary electronic readers are familiar with, and highly value, the simple full text string searching available in word processors, Web browsers, and virtually all e-book reading systems. Pattern matching techniques such as "regular expressions" are sometimes available and can further improve retrieval, as can language specific processing such as identifying the morphological root word (e.g., minus indications of case or number), or matching against a thesaurus entry of semantically equivalent words. But most important, the underlying XML vocabulary can often be exploited, more or less as database fields, to improve retrieval precision, locating strings of characters only when marked as, for example, a *person, place, chemical, disease,* etc.; or only when within a *title, quotation, warning, poem,* etc.

### Currency

Electronic books can be very easily updated. In fact one can imagine how, exploiting daily synchronization by wired or wireless networking, an e-book might be refreshed every single day, or even periodically throughout the day, to reflect changes in its subject matter: mistakes can be corrected, cautions added, statistics updated, new information included.

### Multimedia

Electronic books can present audio and video content with a wide range of applications, from an instruction video in a manual, to pronunciation examples in a language textbook, to newsreels in a history textbook, to avant garde multimedia fiction. Although a great deal of the academic research on high-performance reading focuses on technical manuals and textbooks, the commercial possibilities in mass market and trade books are probably just as great: it may soon be hard to imagine that popular biographies of actors or musicians were ever simply text and still pictures. (See the chapter on multimedia to get a sense of the possibilities.)

### Interactivity and special processing

The availability of computer-based processing during the presentation of e-book content to human reader makes possible a wide variety of useful interactions. For instance, equations can be modified and re-evaluated by the reader, and charts, diagrams, and graphs can be "active," allowing the reader to change

the data, coefficients, formulas, and axes, and see the consequences immediately. Exercises and tutorials can be presented and then automatically analyzed or graded, and the reader directed to the relevant remedial sections. A great deal of research is currently underway in this area, particularly for STM and educational publishing, as features of this sort have an enormous potential to provide unique valuable advantages.

### Accessibility

Paper books present a huge problem for readers with perceptual disabilities. Electronic books, however, can easily make adjustments in type size, color scheme, and user interface. In addition, well-designed content can be easily adapted to high-function audio presentation, complete with cues for structure, hypertext, and other navigation possibilities. See the chapter Accessibility for a discussion of these issues.

# Thinking clearly about e-books

## Navigating the hype

During the years 1999 and 2000 there was a flurry of commercial activity, and trade news, based on the expectation that a commercially significant e-book industry was imminent. There was a sense that the magic combination of improvements in hardware, software, standards, user interfaces, back-end workflow, connectivity, critical mass of content, publisher interest, consumer interest, rights management, etc., was finally here (or at least just around the corner) and that the combination would be explosive: e-books would replace paper books, society would benefit from improved functionality, and those companies in the vanguard of this change would benefit commercially.

To knowledgeable disinterested observers the predictions of the imminent success of the e-book seemed unlikely, given the state of the technology and the general complexities of the publishing industry. But because commitment to unlikely outcomes is at the heart of entrepreneurial success, it also seemed inappropriate for skeptics to do more than express caution and wish everyone the very best of luck. As investments increased, as new companies formed, and old companies launched new projects with greater and greater fanfare, there was increasingly more to lose in backing away from the earlier confident predictions—and so those predictions continued well past the point where it was obvious to almost everyone that they were unjustified. Finally, beginning in late 2001, with the same overwrought drama ("The Death of the Book!") with which the ambitious predictions had been made in the first place, trade journalists began to trumpet the failure of those expectations ("The Death of the *e-Book!*")

The truth is that these theatrics have little to do with the long-term trends in technology and everything to do with how entrepreneurial markets work, what raises investment funding, what promotes individual careers, and what draws attention to stories in the trade and popular press. And, of course, the generally optimistic mood of the mid-1990s had a role to play as well.

What remains true is that the basis for predicting a thriving e-book industry at some point in the near future remains as strong as ever. And, moreover, the empirical evidence, which corroborates that analytic basis, is equally clear: during the same period when the failure of the e-book was being opportunistically trumpeted by the trade journalists, actual e-book sales increased, while paper book sales remained flat or declined.

Making rational predictions about whether, when, or how e-books will emerge as commercially significant requires remaining relentlessly focused on the evidence—and cautious about any but the most well-supported, and least specific, predictions.

One must also avoid "red herrings"—things that are true, but irrelevant, and therefore misleading. Among the red herrings particularly disruptive to reasonable thinking about e-books are:

### The replacement red herring

This is the claim that paper books and paper-based printing will (or will not) be completely replaced by electronic books. Paper books and paper-based printing will probably not "disappear" at any point in the foreseeable future—but that fact, never contested by any serious industry analyst, has *nothing* to do with whether or not electronic publishing will soon be important, and perhaps even more important, than print publishing, let alone with whether e-books in particular will be a commercial success, or how much of a success they will be.

### The bed-beach-bath red herring

According to this red herring, no one will want to read an e-book in bed, in the bathtub, or at the beach. Whether this is so or not, and to what extent it is so, if it is, is irrelevant: bed, bath, and beach reading venues are commercially insignificant and therefore have little to do with the prospects of e-books. Commentators who smugly write that they cannot imagine taking an e-book to bed/bath/beach only advertise their ignorance of both actual reading behavior and the publishing industry—or their willingness to forego useful analysis in favor of an appealing but misleading image. (And, in any event, people do read e-books in bed, taking advantage of their light weight and unobstrusive backlighting.) The bed/bath/beach commentators may defensively claim that this is only intended as a playful allusion to the superior general aesthetics of paper book reading—but that claim itself might be called...

### The lots-of-advantages red herring

According to this red herring, paper books have lasted as long as they have for good reason: they are intuitive to use, pleasant on the eyes, feel nice, don't need electricity, are fairly cheap, never crash, etc. Of course paper books have many advantages, and, indeed, many advantages over e-books—just as e-books have many advantages over paper books. The issue is not whether some technology has some advantages over another, but rather at what point, as technology and habits change, will the *weighing of competing advantages* begin to result in certain sorts of e-books being commercially important in certain publishing markets. Despite the many real advantages of paper-based reading and the many real disadvantages of electronic reading, most people in the developed world spend an enormous and increasing amount of time engaged in electronic reading.

## Why we know that electronic books will happen

There is a broad general agreement among most publishers that electronic books will eventually be a commercially important publishing product. Reading is obviously becoming increasingly electronic (in fact, for many professionals in the developed world, it is mostly electronic) and publishers' data show that e-book sales, although still very small, are, unlike paper book sales, rapidly increasing. Most important, the specific advantages enumerated above remain as important as ever, and the technologies and infrastructure required to realize those advantages continue to mature and advance. While opinions may vary widely on the exact schedule at which e-books will become commercially important, what form they will take, what markets they will appear in first, or what structure the industry will take, few if any publishers doubt that it is only a matter of time before e-books are a significant part of our reading lives.

But since optimistic predictions about e-books have been wrong several times in the past, we might reasonably ask what, specifically, has changed to make this prediction reasonable now? Here are the most important changes:

## Hardware improvements

Storage capacity, processing speed, size, weight, screen resolution, battery life, have all undergone enormous improvement in the last ten years, and they continue to improve. In some cases—storage capacity and processing speed for instance—the improvements are astonishing, literally orders of magnitude. General design, ergonomics, and aesthetics have also substantially improved. In the past, even right up until a year or so ago, many of these things were all cited as negatives; that soon will no longer be the case.

## Software improvements

New carefully designed user interfaces, navigation techniques, information retrieval, annotation capabilities, special processing (e.g., multimedia, interactivity, dynamic diagrams, integrated reference tools) all provide attractive new functionality that is extremely important in the STM and education markets. Particularly important for competing in mass market and trade arenas are recent text rendering techniques that offer a reading experience nearly equal to print.

## Ubiquity of reading devices

Today in the developed world the near ubiquity of laptops, cell phones, and PDAs, most of which are already used for reading of some sort, provide an already existing base of possible candidates as e-book reading devices. In addition there are growing numbers of tablet computers and specialized e-book devices.

## Critical mass of content

Whereas until recently electronic delivery meant converting print material or typesetting files to new digital formats, text creation is now almost wholly digital from the start ("born digital"), and for the most part content is prepared with the anticipation of electronic as well as print delivery. As a result the most commercially important content, new content, is almost always already in electronic form and ready, or nearly ready, for digital delivery.

## Data standards and interoperable tools

The increasing acceptance of SGML/XML element vocabularies for in-house archival formats as well as for interchange has substantially improved the interoperability and functionality of available content and stimulated the development of new software tools and applications for creating, managing, and delivering this content. (To be sure, much content is still created in low-function, noninteroperable desktop publishing and page description formats, but it is only a matter of time before these are either replaced by, or integrated with, SGML/XML-based systems.)

## Culture of electronic reading

Due largely to the emergence of the World Wide Web and widespread e-mail use, the revolution in reading is not something that will happen—it is something that has mostly already happened, at least in the developed world. Obviously in the workplaces of the various professions and bureaucracies, reading and writing at the computer is the dominant form of work, as any tour through a modern office building will immediately reveal. Schools and universities do still account for much nonelectronic reading, but they are experiencing explosive growth in computer-based reading, as the typical college syllabus, on-line and filled with links to on-line reading assignments, demonstrates. Finally, the amount of time that all of us—adults, children, and teenagers—spend at the computer outside of work and school (surfing the Web, reading e-mail, shopping, listening to music, chatting, etc.) is also already very large and growing fast. So

fast, in fact, that parents are now alarmed at the amount of time their children—or their spouses—spend at the computer. Today in 2002, unlike 1992, the special advantages to electronic reading (fast access to many resources, following hypertext links, retrieval by key word, phrase, or more sophisticated means, navigation, integration with other activities, multimedia, etc.) are familiar to, and valued by, almost everyone.

## What we don't know about *how* they will happen

However, while it now seems likely that electronic books will become commercially significant fairly soon, many things about just what this success will look like remain uncertain.

### Form factor

What will be the most popular form factor—PDA-size, book-size, tablet-size devices? Will specialized devices fail to compete with the ubiquity, power, and familiarity of the laptop? Will the special advantages of tablet-sized readers overcome our reluctance to carry yet another device? Will the convenience of PDAs be more important than screen size and resolution? Will electronic paper offering lightweight or roll-out screens solve the size/readability dilemma? Perhaps different devices will be popular for different publishing markets—say tablet devices for magazines, laptops or notebooks for textbooks and STM, PDAs for mass markets? As is obvious from the profusions of different designs already in the market, there is little agreement amongst the experts on this.

### Publishing markets

Where will we see commercially important e-book penetration first: mass market fiction? magazines? textbooks? technical documentation? STM publishing? business documents? reference works? Which markets will resist e-book penetration?

### Industry structure

How will the current ecology of publishers, distributors, composition houses, content developers, software manufacturers, and hardware manufacturers change? Will publishers become distributors? Will distributors become publishers? Will in-house composition become popular again? Will conversion services become unnecessary ... or more important than ever? Who will flourish? Who will be "disintermediated?"

### Business models, intellectual property, digital rights management, and security

Will content be sold or licensed? If the latter, will it be pay-per-view, pay-per-month, or lifetime access? Transferrable or nontransferrable? Sold piece by piece or in large lots, to aggregators? What will be the infrastructure for digital rights management? Will content be encrypted and if so how? (See the chapter Digital Rights Management for a discussion of these issues.)

# The format problem

## Introduction

In addition to the uncertainties described above, individual publishers and the industry as a whole face another family of problems, probably the most serious of all: a proliferation of competing noninteroperable and typically low-function content formats at every stage of the content development and delivery life cycle.

Content formats can be classified along different dimensions. The first and absolutely crucial distinction is the difference between *logical* (or *structured)* formats and *presentational* formats; this distinction is the foundation for all sound reasoning about publishing strategies. In addition, reasoning about production strategies and, particularly, how to take advantage of existing electronic production processes for possible sources of digital content, also involves appreciating the characteristics of *binary* vs. *text* files, and *revisable* vs. *nonrevisable* files. We review these notions below. (For simplicity of exposition we allow ourselves to use the word "format"—and "file" and "content" as well—somewhat ambiguously; there are various fine distinctions to be drawn, but doing so is not necessary here and would complicate the discussion.)

We then present a second independent categorization of formats based on whether they are typically found early or late in the e-book production process, and then, in the case of those found early in the e-book production process, we distinguish those found early or late in the *print* production process. These format categories, representing multiple dimensions, can help publishers new to e-book publishing sort out the immediate complexity that they are faced with when trying to align an e-book production project with already existing print production.

On the basis of this discussion we go on to consider some difficult challenges facing the individual e-book publisher, and the industry, with respect to content formats: coping with the sheer number of relevant formats, both for original content, and for final delivery; supporting functionality and innovation (or "competitive differentiation") without sacrificing interoperability; and leveraging current practices while simultaneously planning for future opportunities.

Next we sketch the general form of the solution to these challenges.

Then, in the following major section of this chapter we present a particular implementation of this solution: the OEB Publication Structure.

# Categories of formats

## Logical vs. presentational

The logical/presentational distinction is now widely recognized as fundamental to the design of workflow systems, to the choice of file formats, to the development of business strategies, and to sound reasoning about the publishing production process generally. This distinction is discussed at length in the chapter on markup and some knowledge of the material presented there is assumed in the remainder of this chapter.

We will, however, reiterate several key observations:

### The logical approach separates structure from presentation

The heart of the logical approach to publishing is to organize the publishing process by identifying and describing what one might call the *logical* (or, alternatively, *structural*) components of the content (e.g., the title, author, chapters, sections, extracts, lists, technical terms, equations, citations, etc.) and then use rules to map presentational features (e.g., font size and style, lineation, pagination, horizontal and vertical spacing) to the logical component *types*—rather than associate processing with each individual component *instance.*

In the case of content being presented only in print form, or digital content primarily designed for traditional linear reading, often only the traditional "editorial" logical components are identified, as only those require typographic distinction. However, where the content is intended to support advanced functionality, many kinds of components (possibly including very fine-grained classification of domain-specific items: for instance, names of persons, places, and countries in a history book; or diseases, organisms, drugs, organs, anatomy, and treatments in a medical book) will be identified, regardless of

whether they will be formatted distinctively, in order to support such things as information retrieval, navigation, linking, special processing, and the like. (In some cases publishing aimed at traditional print output also exploits fine-grained logical markup in order to manage integration of content from various sources, validation, indexing, and reuse.)

#### Descriptive markup implements the logical approach

"Descriptive markup" is used to implement the logical approach. Descriptive markup identifies and often further characterizes instances of logical component types as those instances occur in content. Descriptive markup is contrasted with procedural markup, which indicates how something is to be formatted, not what it *is*.

#### SGML and XML define descriptive markup languages

SGML and XML are "metalanguages" designed to support the creation of rigorous machine-readable definitions of descriptive markup languages and to minimize arbitrary variation among markup languages. XML is a simplified version of SGML, which is now widely used in publishing and information processing. Examples of SGML/XML markup languages are HTML/XHTML, TEI, DocBook, and ISO 12083. (Again, see the chapter on markup for more information on SGML and XML and related standards.)

#### The logical approach yields many advantages for publishing

Publishing systems that are based on a logical approach to content generally provide vastly superior functionality, interoperability, efficiency, and cost effectiveness, compared to systems that are based on a purely presentational approach. It can be easily seen how this approach makes document creation and global changes in formatting simpler, but there are many other opportunities for improved functionality and interoperability as well. (See the chapter on markup for both *why* this is so, and *how* it can be exploited.)

#### Presentation formats are still important

Although publishing systems based on the logical approach are superior in general, logical formats are not necessarily the right choice at every point in the publishing lifecycle—various presentational formats, derived from logical formats, may be superior at stages near final processing for presentation to a human reader.

#### Compromise strategies may be necessary

In addition, the complexity of real-world publishing may require compromise solutions in any case. Conditions creating the need for compromise and decisions to choose suboptimal formats and production processes include: financial constraints; content in legacy formats; installed software; available expertise; difficulty changing production processes; hardware limitations; marketing timeframes, ROI timeframes, and business strategies; and the need for security and digital rights management. Nevertheless, it is the opinion of the authors of this chapter that resisting the development of production systems based on the logical approach (however qualified) typically results in unnecessary costs and loss of business opportunities.

### Revisability and representation

#### Revisable vs. nonrevisable formats

*Revisable* formats are intended to be easily edited to correct or revise content or formatting. Obviously word processing and page composition programs typically create revisable formats. Examples include

native MS Word and QuarkXPress files (both "binary" files, see below) and LaTeX and SGML/XML files (both "text" files, see below).

*Nonrevisable* formats cannot be easily edited. Obviously nonrevisable formats tend to be output oriented. Examples include raster image files and device-specific typesetting files such as Linotron 202 data ("binary" files), and page description files such as Adobe PostScript and PDF (both "text" files).

### Text vs. binary formats

*Text* formats represent their data with a standard seven- or eight-bit byte character encoding. An example is ASCII, which uses one byte per character; another is Unicode, which may use more than one byte per character. Although the use of the word "text" in this sense is common in information processing, it can be confusing: the term does not refer to the linguistic text of the document, nor is it to be contrasted with graphical or other media such as audio or video. "Text" in this sense simply means that the information represented—whether that information be formatting instructions, vector graphics, natural language prose, indexing locations, metadata, or hyperlinks—is encoded via a standard seven- or eight-bit byte character encoding. ASCII text files can be easily read by general-purpose editors and Unicode text files by editors that process multiple-byte characters. Examples of text formats are SGML/XML, Microsoft's RTF, Adobe FrameMaker's MIF, and Adobe's PostScript and PDF. Obviously text files vary very widely in their application and nature: some of them are revisable, some not; some are logical, others presentational.

*Binary* formats do not encode their data in a standard, easily read seven- or eight-bit character encoding such as ASCII or Unicode, but instead use an application-specific encoding. Binary formats cannot be usefully viewed in a general purpose text editor and are typically processible only by the creating application, applications from the vendor's suite, or specially created "black-box" conversion applications. Both the authoring software in common use for creating original content and the composition or page layout software most commonly used to lay out content for typesetting typically save their content in binary formats (e.g., native Microsoft Word files or native QuarkXPress files).

## Formats up- and downstream in e-book production

It is natural to think of e-book production as a process that starts "upstream" with the original creation or identification of content, and then continues, "downstream," with the transformation of that content into various formats, including those that represent the page layout and rendering of the content.

The riverine metaphor is an oversimplification, of course. For one thing, there are iterative cycles of corrections and revisions that result in content flowing upstream as well as down—and the authors of this chapter would be the last to suggest downplaying this cycle: this information, and the techniques for creating and transmitting it, represent process knowledge that is fundamental to the effectiveness of an organization's workflow. Moreover, the propagation of corrections and revisions upstream, which is necessary in order to create the ideal "many-products-and-formats-from-a-single-source" production strategy described below, turns out to be a challenging process to manage efficiently.

We also note that most format categories can be found almost anywhere in the production process, or at least within a broad interval, so when we say that certain formats are found upstream, or downstream, we mean typically or for the most part.

Finally, and most important, we are taking the perspective of someone trying to align or integrate an e-book production with an already existing print production process; this means that formats that are *down*stream from the perspective of print production process may still be *up*stream from the perspective of e-book workflow. This perspective reflects the practical fact that the e-book publisher is typically not designing an ideal production process from the ground up and independently of any other constraints or activities, but is more likely making a connection with an already existing publishing workflow.

However, even if the production process is built from scratch to reflect current best practice, or an existing production process is restructured and rationalized, the metaphor still works. An ideal publishing production strategy, based on accommodating multiple content formats as inputs and then creating multiple delivery formats from a common interchange format (as described further below), still yields the topography of a river: one that has both tributaries and a delta.

In the section that follows, we organize our discussion of the various formats found in the production process in terms of where in the process they are most likely to be found.

Formats found upstream in the e-book production process and upstream in the print production process:

Revisable binary formats and their derived revisable text formats

Revisable text formats (general purpose)

Formats found upstream in the e-book production process and downstream in the print production process:

Revisable binary formats and their derived revisable text formats

Nonrevisable binary formats

Nonrevisable text formats

Formats found downstream in the e-book production process:

Nonrevisable binary formats

Nonrevisable text formats

Revisable text formats

## Formats found upstream in the e-book production process and upstream in the print production process

### Revisable binary formats and their derived revisable text formats

There are a number of problems at the outset with using revisable binary files, such as those created by word processing or page composition software, as a source of digital content. Binary formats cannot usefully be edited with plain text editors. Most binary revisable files are in proprietary formats that have no public documentation and can usually only be used with a single software application or a suite of products from a single vendor. It is difficult to convert binary files into other formats unless this conversion has been anticipated and provided for by the software vendor.

Generally it is only possible to exploit binary formats if they can be converted into a *revisable text* format. However, the usefulness of the resulting text format will depend further on whether the original revisable binary format contained structural logical information, whether this structural information was preserved in the derived text format, and whether the revisable text format itself has a documented encoding that can be easily manipulated for further transformations.

Unfortunately, most text processors and page layout programs in common use fail to provide adequate support for the logical approach to text creation and publishing, and few create output in nonproprietary easy-to-transform representation schemes like SGML/XML. The text files produced by these programs are in a vendor-developed encoding system specifically associated with the original application (or a suite of the vendor's applications), and optimized for the support of interchange. A familiar example from general text processing is Microsoft's Rich Text Format (RTF), and one from publishing and technical documentation is Framemaker's MIF format. Although a text-based format (if publicly documented) is obviously easier to exploit than an undocumented binary format, these proprietary revisable text formats vary widely in their suitability as a source of content for e-book publishing. It may be possible to extract

some structural information (if it is present) from vendor-specific text formats, but it will typically require expensive conversion based on ad hoc programming.

### Revisable text formats (general purpose)

Many revisable text formats are not closely associated with a particular software application or a particular vendor either. Well-known examples of a general purpose revisable text format in the area of education and STM publishing is TeX and its derivative LaTeX. Others are Scribe and troff. One that is associated with a particular vendor, but is still general purpose, is IBM DCF/GML.

As always the critical factor in determining the suitability of revisable text format for e-book content is whether these formats include information about the logical structure. Content in markup systems where descriptive markup typically predominates, like LaTeX, IBM DCF/GML, and Scribe are usually very good candidates for converting into e-book formats either directly, or, better, through an SGML/XML interchange language. Only inspection and analysis will determine the actual extent to which structural information has been encoded. Even rigidly structured well-thought-out markup languages can be circumvented by careless or indifferent content developers.

A text-based revisable format that is particularly important to e-book publishing is, of course, SGML/XML, and the various specific SGML/XML vocabularies such as TEI, DocBook, NewsML, XHTML, and the ISO 12083 derivatives common in STM journal publishing, as well as the many other local or in-house SGML/XML formats. As noted above, SGML/XML implements the "logical" or "structured" approach to publishing by providing a rigorous but flexible metalanguage for defining descriptive markup languages, reflecting throughout its design and philosophy publishing best practice and fundamental scientific principles of information organization. This provides all the advantages of functionality and interoperability described above, as well as improving ease of conversion (using techniques like XSLT or ad hoc programming) into a variety of other formats—usually at much less cost than the comparable transformations from vendor-specific formats. In addition, unlike the proprietary binary revisable formats and their derivative text formats, many SGML/XML vocabularies were often designed specifically to support electronic as well as print publishing. The identification of SGML/XML content upstream is valuable to the e-book publisher for several reasons, but primary among them is the possibility of creating downstream formats relatively easily. For the e-book publisher looking for sources of digital content, the discovery of material in a well-designed SGML/XML vocabulary, used with consistency, care, and discipline, is priceless.

Some of the word processing and page layout applications in common use can produce SGML/XML output, instead of, or as well as, both native binary files and text files in a vendor-specific encoding. This can be a promising source of content for the reasons noted above. Unfortunately—and this is a matter of considerable practical importance in the current publishing environment—not all SGML/XML content produced from common word processing and page layout software lives up to its promise. First, the structural information actually has to be created in the first place. If it is not there, whether because the software application does not provide the user with an effective environment for easily creating structured documents, or because the user did not use the software in a consistent disciplined way, then this information won't be present in the SGML/XML output either.

Publishers must beware of exaggerated claims by software manufacturers who will make it sound like creating useful SGML/XML output from their applications is just a matter of choosing SGML/XML from the "Save As" menu. This makes as much sense as choosing "best-seller" from the menu and expecting to publish a best-seller. The result may be SGML/XML in appearance and even in syntax, but will be little better than a typical presentation-oriented vendor-specific format unless the software application is specifically designed to support the logical approach and create structured documents, and the software is used in a consistent and disciplined way, with the intention of creating content in a structured logical format. For the same reason, e-book publishers must be cautious when obtaining SGML/XML content

from other publishers or from conversion houses. Conforming to the syntactical conventions of SGML and XML is easy enough, but if the design or the execution is inadequate the cost and functionality benefits will not be achieved. Publishers new to electronic publishing should seek an independent analysis of the quality of the encoding they will receive before they make a major purchase or invest in conversion. Good professional analysis of SGML/XML content is inexpensive and easy to arrange.

## Formats found upstream in the e-book production process and downstream in the print production process

### Revisable binary formats and their derived revisable text formats

In most commercial publishing, text creation and development takes place initially in word processing applications, and then the content is transferred to page composition programs for final page layout, although in some cases the same applications are used throughout the process, from creation to printing. In any event, downstream revisable binary formats have undergone final preparation of typography, graphics, indexing, lineation, and pagination and such. These formats often seem like attractive sources of content. For one thing, they are likely to contain more corrections, revisions, and other useful value-added improvements than the various earlier upstream revisable versions. In addition, they are often easier to get a hold of, being more likely to be retained by the typesetter than the original revisable files are likely to be retained by the publisher or compositor. (Although often it is discovered that a supposedly final typesetting tape doesn't match the printed book—because corrections were reset and added in pasteup or even in film at the printer.)

Unfortunately the transition from the development applications (like word processors and SGML/XML editors) to page layout applications often results in the systematic replacement of structural information, now presumably no longer needed, with formatting codes, and other specialized encoding intended only to support formatting, and not development, conversion, or any other use. The conversion of such formats (even if a text version is available) into other formats is almost never financially practical, and often not possible at all. These formats are typically highly specific to a particular application, and difficult to parse and transform. But most important, whatever structural information was available in the original revisable versions is now lost, having been replaced by application-specific processing codes. And it is this structural information that is needed for efficient creation of either new application-specific processing codes, or general-purpose interchange format. Regenerating structural information about the logical components of the document from application-specific processing codes, even when that information existed in the original revisable versions, is rarely financially feasible, even apart from the binary format and the undocumented, often ad hoc, syntax of the data: the relationships between codes and components are many-to-one (something can be bold or indented for many different reasons) and conditional on the now-missing original contexts. While it is true that information is added to content representation as it moves downstream, information is also removed, and by the time this representation reaches typesetting format much critical information is gone and cannot be recovered.

### Nonrevisable binary formats

Downstream revisable formats may or may not retain structural encoding, and may or may not produce tractable text files, but downstream nonrevisable binary formats, such as device-specific typesetting files, never retain structural information, and rarely can easily produce a revisable text version of their content. They cannot be used for repurposeable digital content.

### Nonrevisable text formats

There are two downstream nonrevisable text formats of particular interest to the e-book publishers as sources of digital content—Adobe PostScript and Adobe PDF. Adobe PostScript is a powerful

programming language for creating typeset pages with vector graphics and high-quality fonts on all-points addressable devices such as laser printers. PostScript was a key part of the emergence of desktop publishing in the mid-1980s, and today almost all word processing and page layout software creates PostScript output (a mixture of PostScript commands and data), which is processed by PostScript printers to produced formatted pages. Adobe PDF (Portable Document Format) is also a page description language; PDF files are typically produced directly from PostScript and are more suitable for electronic distribution. Publishers today usually produce and archive PDF files as a matter of course during the print production process, and so PDF (and PostScript) are downstream in the print production process, although upstream from the point of view of an e-book publishing project that is being undertaken within the context of an existing print publishing workflow.

PDF and PostScript are extremely attractive as possible sources of content: they are today routinely created and archived by almost all publishers, typically contain all previous corrections and value-added information (the contemporary full-page makeup systems that produce PostScript rarely rely on mechanical pasteup to make final corrections—instead the entire publication is reformatted with each change), and their well-defined publicly documented text format might seem, at least at first glance, well-suited for transformation.

Unfortunately, PostScript and PDF files contain little of the original structural information, even when that information existed in the revisable formats, and the formatting information they do contain cannot be used to regenerate the missing structural information. What PostScript and PDF files contain is information needed to reproduce the *visual,* rather than *logical,* format of the page—the fonts, their sizes and position, colors and graphics, and so forth. In some cases the alphabetic text can be extracted, but this is typically a crude process and provides at best only a slight advantage over new data entry. It is true that newer versions of PDF can contain some structural information, but is unclear at this point whether this information, even if added by the original content producers, will be adequate to make PDF a practical source of digital content. (All this is not to say that PDF is not a reasonable e-book *device format,* especially when the intent is to reproduce the appearance of the original typeset pages—only that it is not a reasonable source of digital content for creating revisable formats, a general-purpose interchange format, or for creating non-PDF device formats.)

## Formats found downstream in the e-book production process

The vast number of publishing formats, many proprietary, rigid, and noninteroperable, poses a major problem for e-book publishers trying to design publishing strategies that are integrated with existing print-oriented workflows. But to make matters worse, entirely new content formats have been developed, and continue to be developed, specifically for the e-book devices themselves. Publishers of e-books must not only cope with diverse multiple formats as sources of digital content, they must produce a variety of different formats if they intend to make their e-books available on the many e-book reading devices already in existence.

### Nonrevisable binary formats

These new formats include most prominently Microsoft's ".lit," (for the Microsoft Reader), Gemstar-TV Guide's ".rb" (for the RCA e-book), and Palm's ".pb" —following the usual practice of indicating these formats by their characteristic file extensions. These are all nonrevisable binary file formats designed specifically for e-books and specifically for the e-book reading software manufactured by each of these companies. They are all proprietary and all noninteroperable. Apart from the business strategies that motivate the development of any proprietary content standard, there are several other technological and commercial motivations that are frequently claimed for binary e-book formats:

binary formats are faster and more compact

binary formats can be a vehicle for delivering innovation and functionality that could not be supported by directly processing OEBPS XML content

whether or not an innovation could be provided by directly processing OEBPS XML content, binary formats can help protect innovations and special functionality, providing competitive differentiation

binary formats can more easily both secure content and provide digital rights management

#### Nonrevisable text formats

The only nonrevisable text format that has any significant presence in e-book publishing is Adobe PDF. But this content format, particularly in the version used with the Adobe Acrobat eBook Reader, is an extremely important e-book format.

#### Revisable text formats

Some Reading Devices can process and present HTML files and ASCII text files without markup. However, the revisable text device format that is important to e-book publishers is OEBPS itself. Some e-book reading software (e.g., the readers produced by ION Systems and Globalmentor) process OEBPS directly, without conversion into any other format.

## Format-related challenges

### Functionality, innovation, and competitive differentiation vs. interoperability

In choosing a content format for publishing, two desiderata immediately present themselves as requirements, and just as immediately appear to be impossible to achieve simultaneously:

1. *Support for interoperability.* Content should be readable on multiple reading systems, and reading systems should be able to read many different kinds of content. Obviously this is valuable for business-to-business relationships in publishing, but it is even more critical for consumer confidence, and for making content as widely, and as reliably, available as possible. The experience of 1980s word processing—when most software created noninteroperable formats that couldn't be edited by software from other vendors, processed by third-party tools, or often even printed on arbitrary printers—must not be repeated in the e-book industry, where e-book interoperability is constantly compared to the nearly complete and transparent interoperability of paper books.
2. *Support for innovation.* Content developers and reading systems both should be able to provide new features, providing new value to customers, and "competitive differentiation" to businesses. The full realization of the promises of electronic reading will only be achieved over time, as advanced features prototyped in experimental systems are perfected and made commercially available. Format requirements must not prevent this diffusion of innovations.

But how can these two things be possible simultaneously? If in order to ensure interoperability a single e-book format is chosen as the industry standard, then that format will have a specific combination of capabilities. It will, for instance, support certain formatting features, certain media types, certain graphical features, a certain level of interactivity, and so on. Content that conforms to this format will be restricted to *only* these features; software that conforms will process these features, but possibly no others. Content providers who innovate beyond these features in their e-book content cannot be assured that their innovations will be processed appropriately and reading system developers cannot be assured that they can process this new content—unless there was an arranged agreement between the content innovator and the reading system developer. In these circumstances interoperability will quickly evaporate, and business strategies based on format and feature "wars" will further erode consumer and industry confidence, threatening the possibility of developing a thriving e-book industry.

### Functionality vs. current reality

Real technological change always takes place in *medias res* and pretending that this is not so—pretending, that is, that we can develop plans to improve our current circumstance simply by figuring out what things would be like in an ideal world—rarely succeeds. Today the publishing industry possesses much content in various unpromising legacy formats, production systems unsuited for electronic publishing (unsuited, some would argue, even for efficient print publishing), staff with particular skills, certain financial vulnerabilities, and existing business strategies. Unless it had an enormous amount of capital, and could survive a lengthy period before the investment was recovered, a company that attempted to change everything at once and implement the ideal system would be unlikely to succeed commercially. This is a problem for individual companies, but it is also a problem for the industry: how can we accommodate, or even better, take advantage of, current practices, and at the same time prepare the industry to take best advantage of future opportunities?

### The problems of format conversion

E-book publishers must decide what format(s) to choose for their e-books: HTML, XML, .lit, .rb, .pb, .wap, PDF, and so on. Each has advantages and disadvantages. But most disturbingly, each appears to limit the market to just those devices that use that format.

One solution to this problem is to settle on a single format, at least for delivery. But this is unpromising. For reasons of business strategy, differing technology commitments, and differing views about what approach would be best, it is unlikely that publishers and reading system manufacturers would agree on a single format. And even if they did, this would quite likely have a stultifying effect on progress and innovation in e-book functionality.

Another possible solution is to allow a thousand flowers to bloom, and assume that conversion software will be developed, as needed, to convert one format into another, at least insofar as such conversion is possible. But this solution too is problematic, as some simple combinatoric arithmetic will show. It would certainly be reassuring to have a conversion program that would convert every content format into every other format, but as can be easily deduced by the fundamental principle of counting, that would require $(n^2 - n)$ conversion programs for $n$ formats: for the seven formats mentioned above, forty-two individual conversion methods. This number grows exponentially as the number of formats grows.

Of course complete format interoperability, even if possible (and some conversions are not practical under any circumstances), is more than is required—we are only really concerned with converting certain common, and reasonably convertible, upstream formats into downstream formats. But although the combinatorics here are better, they are still prohibitive from the practical point of view. The number of conversion programs needed to convert every relevant upstream format into an e-book format is the number of input formats multiplied by the number of output formats. Unlike global conversion, this function grows linearly, but it still grows fast: for instance, if there are twenty upstream input formats and ten downstream e-book formats, there will be 200 required conversion applications, and each additional input format will require ten more conversion routines. To have a sense of what this really means in practice remember that formats are frequently updated, and each new version will typically require updating the relevant conversion applications as well.

## Solving the format problem

These are the problems that the OeBF Open eBook Publication Structure was specifically designed to address. The basic idea is to define an interchange format that could mediate between publishers' formats and the various reading device formats. This would allow publishers to concentrate just on converting their in-house formats into the interchange format, perfecting this single conversion routine.

The single interchange format would then be the common coin of electronic publishing. The next step, converting from the interchange format into the final reading device format, could be performed by the reading system software or by distributors or by other partners. In any case the steps in the process are dramatically reduced in number: whereas the number of conversion routines necessary to convert publisher formats to device formats is inputs *multiplied* by outputs, the number needed for conversion mediated by an interchange format is inputs *added* to outputs.

Many of these input formats are proprietary and most are noninteroperable—that is, software that reads one format usually won't read any of the others. In addition, the nonrevisable formats are not only difficult to edit and update but for the most part difficult to adapt to different reading systems and delivery modalities. Almost all of these formats are specifically designed for supporting print publishing, and even when they nominally support electronic publishing, they are often inadequate for realizing the full range of functionality (as described above) anticipated for high-performance electronic reading. It is therefore difficult for publishers to get their content into a format appropriate for any sort of electronic publishing, and particularly difficult to get content into a format suitable for high-performance e-book reading. From the industry perspective, these problems mean that it will be hard to develop the critical mass of interoperable high-function content needed to create a thriving e-book industry.

Publishers are faced with worrisome questions: will a single proprietary device format eventually become dominant, and if so, which one? Or will there be a number of competing formats? Perhaps the number of formats will continue to increase, driven by efforts to innovate. From the publisher's perspective the situation is discouraging: the cost of converting their content to each possible e-book device is enormous and makes a commercially viable e-book venture look financially unpromising.

From the perspective of the industry as a whole the situation looks even worse. Even ignoring problems of individual idiosyncrasies, the number of qualitatively different conversion routines needed to convert every original content format into each device format is a geometric function that rises fast as new formats are added at both ends: the number of required routines is the number of input formats multiplied by the number of output formats.

Equally important is the perception of consumers: if consumers believe that buying a particular device will limit what they can read to books in a particular format, or that their purchased content will become out-of-date and unreadable when they buy new devices—obviously the consumer confidence needed to build an industry is not going to be there.

# The OeBF Open eBook Publication Structure

This section assumes a general knowledge of basic XML concepts (such as *DTD, document instance, validity, well-formedness, element, attribute, attribute value,* and *entity*) and some rough familiarity with the general nature and purpose of XML-related specifications such as XHTML and CSS. For more information on XML, and XML-related standards, see the chapter Markup: XML and Related Technologies.

## History

### The development of OEBPS 1.0

The Open eBook Initiative was announced at the U.S. National Institute of Standards and Technology's first *Electronic Book* conference in October 1998. Following that conference, three companies (NuvoMedia, Microsoft, and SoftBook Press) that were manufacturing, or had plans to manufacture, e-book reading systems wrote a draft proposal to initiate an industry-wide discussion. In January 1999, at a meeting widely attended by publishing and software manufacturing representatives, the Open eBook

Initiative was informally organized and an "Authoring Group" was formed and charged with the development of the specification. Victor McCrary, a NIST Technical Manager who had been encouraging and supporting the development of the OEB Initiative, became "Facilitator" of the Authoring Group.

Throughout the spring and early summer of 1999 the specification was substantially expanded and refined by the Authoring Group. In September the specification, now titled *The OEB Publication Structure 1.0* (OEBPS 1.0), was approved by the members of the OEB Initiative and released. For its work the OEBPS Authoring Group received a letter of commendation from the U.S. Department of Commerce.

## The formation of the OeBF

In January 2000, the informal Open eBook Initiative became formally organized as the Open eBook Forum, a nonprofit association of publishing industry stakeholders, with this stated goal:

> … to establish common specifications for electronic book systems, applications and products that will benefit creators of content, makers of reading systems and, most important, consumers, helping to catalyze the adoption of electronic books; to encourage the broad acceptance of these specifications on a worldwide basis among members of the Forum, related industries and the public; and to increase awareness and acceptance of the emerging electronic publishing industry.
>
> http://www.openebook.org/aboutOeBF.htm

OeBF members include major publishers, hardware manufacturers, software manufacturers, distributors, service providers such as conversion and composition companies, key trade associations, public sector agencies, public interest organizations, and other related organizations including individual universities and libraries. It has broad support throughout the publishing and publishing software industry and strong collaborative relationships with key trade organizations, such as the American Association of Publishers and the American Library Association.

In addition to the Publication Structure Working Group (see below), the OeBF now has active working groups in the areas of Metadata and Identifiers and Digital Rights Management and Special Interest Groups in the areas of Business Development, Education, and Accessibility, and well as a systematic infrastructure for collecting and analyzing industry needs, and an architecture group (the Systems Working Group) for ensuring that the various technical products are coordinated and meet identified requirements.

## Formation of the Publication Structure Working Group (PSWG)

In May of 2000 at the first formal meeting of the newly reorganized Open eBook Forum the charter of the OeBF's first Working Group, the Publication Structure Working Group (PSWG), was accepted by the OeBF membership and assigned the mission of "maintaining and advancing" the Publication Structure.

### Officers

The original officers of the PSWG, and their affiliation at the time, were, Chair: Allen Renear (University of Illinois, Urbana Champaign); Vice Chair: Garth Conboy (Gemstar-TV Guide); Scribe: Dorothea Salo (Impressions Book and Journal Services); Maintenance Chair: Garret Wilson (GlobalMentor); Development Co-Chairs: Gene Golvchinsky (FX/PAL), Jerry Dunietz (Microsoft).

In the summer of 2001, Jon Noring became Maintenance Chair and in the spring of 2002, as work shifted from OEBPS 1.2 to OEBPS 2.0, Jerry Dunietz (Microsoft) became Vice Chair, Garret Wilson became a Development Co-Chair, and Dorothea Salo (then at the University of Wisconsin) stepped down as Scribe.

### Composition and current activities

The membership of the Publication Structure Working Group includes representatives of all major electronic publishing constituencies, including public interest groups, as can be seen from the contributor lists included in the standard. The PSWG carries out its work through weekly conference calls, e-mail discussion lists, and periodic face-to-face meetings. Membership in the PSWG is open to all members of OeBF, Principal and Associate, as well as to a limited number of "invited experts" and its records and minutes are available to OeBF members and to invited experts. Drafts of new OEBPS versions are widely circulated to the general public for comments prior to their submission to the OeBF—all comments, whether from OeBF members or others, receive written responses from the PSWG.

### History of OEBPS releases and adoption

**September 1999:** OEBPS 1.0 is released.

**Fall 2000:** A year after its initial release, OEBPS is widely used in the publishing industry and has been incorporated into a number of software tools for creating device formats. An OEBPS conformance validator developed by the Brown University Scholarly Technology Group (STG) is made publicly available by STG and NuvoMedia.

**Spring 2001:** The OEBPS "package file" system for organizing publication content is adopted by the DAISY Consortium's Talking Book standard, now NISO Z39.86. For this collaboration OeBF and DAISY jointly received an award from the International Coalition of Access Engineers and Specialists, recognizing "significant innovative technical contributions to the access engineering profession."

**June 2001:** OeBF releases OEBPS 1.0.1, a "maintenance release" that corrected any errors or inconsistencies that had been reported during the first year and a half of use.

**Spring 2002:** Most commercial e-book publishing, other than that which goes directly from non-XML composition formats to PDF, is now based on OEBPS.

**September 2002:** OEBPS 1.2 is released, with improved support for formatting and presentation.

**Fall 2002:** Work continues on OEBPS 2.0, a major release with extensive new functionality, particularly in the areas of navigation and linking, internationalization, and metadata. Many of the new navigation and linking features, which are already completed, were developed in collaboration with members of the DAISY Consortium's Talking Book standard project and are identical with those of NISO Z39.86.

## Purpose and nature of OEBPS

### Official Purpose and Scope

The Open eBook Forum Publication Structure itself is perhaps best introduced by first considering its goals and objectives. The specification presents the official Purpose and Scope of OEBPS in section 1.1:

> In order for electronic-book technology to achieve widespread success in the marketplace, Reading Systems must have convenient access to a large number and variety of titles. The Open eBook Publication Structure (OEBPS) is a specification for representing the content of electronic books. Specifically:
>
> The specification is intended to give content providers (e.g., publishers, authors, and others who have content to be displayed) and tool providers minimal and common guidelines which ensure fidelity, accuracy, accessibility, and adequate presentation of electronic content over various electronic book platforms.
>
> The specification seeks to reflect established content format standards.

> The goal of this specification is to define a standard means of content description for use by purveyors of electronic books (publishers, agents, authors et al.) allowing such content to be provided to multiple Reading Systems.

## Empirically identified objectives

The official goals articulated in section 1.1 are, at best, rather terse, particularly apart from the context and framework within which they were seen by the authors of the specification. We supplement them here with an expanded list obtained by generalizing from the observed activities of the working group.

Empirically speaking then, the following are the specific objectives that actually guided the decisions, priorities, and reasoning of the OEBPS working group.

### Support consumer confidence in performance and interoperability of devices and books

If consumers are not confident that the reading devices they are considering purchasing will reliably present the e-books they already have or will soon acquire, then consumers will not purchase those devices. And, similarly, if consumers are not confident that the e-books they are considering purchasing can be processed, and presented well, on their current devices, or on a device they might acquire later, they will be reluctant to purchase those e-books. The problem is the same in either case (purchasing e-books or purchasing reading devices), and failure to address it would be absolutely lethal to the development of an e-book industry. Consequently supporting consumer confidence in both performance and interoperability was a high priority in the development of the specification.

However, it is important to recognize that it was not an OEBPS objective to ensure that a single representation of content (e.g., a data file) would itself be processible on any reading device. The objective was rather, as will be explained in detail later, to ensure that a single representation of content was processible by any OEBPS conformant *Reading System*—and therefore, ultimately, the content would be presentable on any Reading Device that was part of a conformant Reading System, yielding the functional interoperability described above.

Device-level interoperability is indeed desired by many members of the e-book community, and OEBPS was therefore designed to make device-level interoperability *possible*—itself a major achievement. But *requiring* device-to-device interoperability of data files is not consistent with the range of commercial e-book publishing strategies already underway. Fortunately, interoperability of content at the Reading *System* level can provide much, if not most, of the functional interoperability desired by consumers.

### Create a critical mass of content

A successful e-book industry requires that a large amount of valuable content become available as fast, as easily, and as inexpensively as possible, and that the continuing production of e-books is financially sustainable. Consumers will not purchase e-books they are not interested in reading and they will not purchase e-book devices unless they believe that there is, and will continue to be, a large quantity of e-books.

### Limit burden on content providers

Content providers, such as publishers, must be able to create e-book content easily, whether by converting existing content to e-book formats or by the creation of new content. Therefore wherever possible the OEBPS makes it possible to exploit already existing content formats, production tools, and staff expertise, and, in general, to support simple manageable production strategies for e-book publishing.

### Support content provider need for reliable high-quality performance

Content providers require predictable high-quality reading system performance; they need to be assured of reading systems' rendering capabilities so that they can be confident that their books would be faithfully presented exactly as they are designed to be. The current situation on the Web, where browsers vary widely in their rendering behavior, is completely unacceptable in high-quality commercial publishing.

### Limit burden on developers of reading software

At the same time, the software companies developing electronic book reading systems need manageable processing requirements. Many current e-book devices still have significant limitations on processing speed, memory, screen size, resolution, and color, and the software engineering required to implement advanced formatting and browsing systems under those limitations is difficult. It is not currently possible, for instance, to completely implement CSS2 in a typical handheld reading device. In addition, functionality that may be possible from an engineering perspective may have prohibitive business consequences, such as cost or development time frame. A content format must therefore constrain processing and presentation requirements to those that can be reliably achieved in order to limit the engineering burden on software developers as well as provide predictable performance for content developers.

### Support distinctive new functionality

Digital documents and computing resources provide extraordinary new possibilities for reading and writing: enhanced navigation, retrieval, annotation, updating and currency, multimedia and interaction, and so on. Moreover, these features are familiar to consumers from their experiences on the World Wide Web. E-books must, as soon as possible, deliver such new features in order to justify their claim to provide superior reading functionality. (But they must do so without diminishing interoperability or reliability, or placing excessive demands on software manufacturers.)

### Maintain equitable opportunities for competitive differentiation

Chaotic competition based on proprietary formats would delay or prevent the development of e-books, reducing interoperability, consumer confidence, and, ultimately, functionality. But completely removing the opportunity for competitive development and innovation/differentiation would be equally damaging to the long-term social and commercial value of the e-book industry, discouraging new entrants and limiting innovation and functionality. The specification must therefore somehow support innovation and competitive differentiation without eroding functionality. In particular, competition must somehow be channeled into areas where it would benefit the industry, and society, the most, in the long run, and away from areas where it would be an obstacle to the development of social and commercial value.

### Position industry practices to evolve with emerging standards

Alignment with the trends and emerging standards in publishing, and in information processing in general, is, of course, critical. Even though efforts must be made to leverage existing practices and provide a content format that has immediate value, this format must be, as much as possible, consistent with emerging trends and standards, so that e-book publishing practices can evolve with publishing practices and standards in general.

### Provide an aesthetically satisfying reading experience

Although many categories of books and documents benefit immediately from enhanced navigation, searching, annotation, and interaction, for others—such as mass market romance novels, techno-thrillers, and bestseller fiction—aesthetic satisfaction and the recreation of the traditional sense of being "lost in

the book" is more important. In all cases, providing a satisfying reading experience, appropriate to the genre of text, is important for the success of e-books.

#### Support other languages and writing systems

Both the opportunities of the global marketplace and the moral requirement to ensure that all cultures have equitable access to the emerging world of electronic reading require that an e-book content format must support the world's languages and writing systems.

#### Support access by readers with disabilities

People with perceptual disabilities must not be excluded from participating in the social and cultural world as that world becomes increasing electronic. Access to books by people with disabilities should not only not be impeded by an e-book content format, but that format should take advantage of the powerful resources of digital information and computer processing to improve and enhance that access, ensuring full participation in contemporary culture and society.

#### Have an *immediate* and *direct* impact on the creation of a flourishing e-book industry

For a thriving electronic book industry to emerge, sound content standards need to be made available and updated in a timely fashion. If the needed standards are not available precisely when they are needed, at critical points in the development of the industry, then that development may be delayed or halted. Or, perhaps worst of all, it may proceed, but with suboptimal strategies that will prevent, perhaps indefinitely, the emergence of more functional and valuable technologies and products. And, in addition to timeliness, strategies for supporting the development of the industry also must avoid extensive dependencies on other projects or initiatives, and must not rely on long or complex chains of effect.

#### An overarching secondary goal: reconcile conflicting primary goals

A glance at the goals listed above shows that many of them pull in different directions (e.g., high-function faithful rendering vs. manageable software requirements), and a number reflect the two fundamental opposed objectives mentioned earlier: interoperability vs. functionality and innovation; and exploiting current practices vs. preparing for future developments. The key challenge for OEBPS was to either develop innovative solutions that would reconcile these oppositions, and to make judicious decisions in weighing the tradeoffs of competing strategies. Understanding these conflicts will put many of the features of the OEBPS, both current and future, in perspective.

### General nature of OEBPS: an application of already existing standards

One might reasonably ask why, when so many other standards already exist, is it necessary to develop yet another? The answer is important, and essential to understanding the fundamental nature of the OEBPS.

There do indeed already exist many information processing standards relevant to electronic publishing, for instance XML, XHTML, CSS, XLink/XPointer, XML Schema, RDF, Dublin Core, Unicode, image formats, sound and video formats, and so on. There are in fact hundreds of relevant standards, produced by large standards organizations such as IEEE, IETF, W3C, ISO, and NISO. These standards are typically powerful, detailed, expressive, carefully crafted, and based on sound principles of information processing. Each covers in great detail a very wide range of needs in its particular area, and, considered collectively, they together include all, or nearly all, aspects of electronic publishing.

However, to provide genuine practical solutions to the various problems described earlier in this chapter it is obviously not helpful to simply gesture toward the hundreds of existing relevant "vertical" standards, however complete and soundly developed they may be. What is necessary is that all players know exactly which standards are in fact being used, and moreover, they must agree on how much of each standard

must be supported—for it is impossible to expect that software manufacturers working within specific hardware and processing constraints, as well as practical engineering demands, always implement all of a selected standard. Practical limitations, and the need for a high-level of predictable capability and behavior, require selection and further specific constraints.

Ironically it is precisely the features that make a specific standard a good one—expressive power, completeness, and flexibility—that can make specific standards difficult to use, without further constraint, in practical commercial contexts.

Consequently, the practical development and use of electronic publishing content requires not just the existence of specialized individual standards, however sound, but it also requires a specific detailed agreement on a coordinated and constrained application of a particular selection from the available relevant individual standards.

### Summary: the purpose and nature of OEBPS

Reflecting on both the official statement of purpose and scope, and the empirically identified objectives, we would, in a sentence, summarize the purpose of OEBPS this way:

> The principal goal of the OEBPS is to contribute to the development of a commercially and socially valuable electronic publishing industry by providing a common format for digital content.

From the perspective of the industry as a whole, having such a single common format will, by improving the interoperability and functionality of both content and reading systems, increase consumer confidence and satisfaction, lower overall development and processing costs, and support innovation and competitive differentiation—all things that are needed to develop a thriving, socially valuable e-book industry.

From the perspective of the individual publisher, there are similar immediate benefits in interoperability, and in functionality of their products, opportunities for more efficient relationships with partners, and improvement in efficiency of production processes, particularly in the areas of integration with existing in-house workflows and for supporting multiple products and multiple delivery modalities.

The specific "empirically identified" objectives listed above amount to a statement of the various *functional requirements* that must, given the actual circumstances of contemporary electronic publishing, be met by the specification in order for it to achieve its goal.

In closing this discussion of nature and purpose we note that although OEBPS is rigorously based throughout on fundamental principles of information processing, and is carefully positioned to evolve with emerging trends and standards, it is also, in every aspect, and as a matter of philosophy, a *practical* solution to *current* problems in electronic publishing. Obviously this is the rationale behind OEBPS's basic nature as a comprehensive specification for applying existing standards. In addition, in all of its design decisions and in all weighing of tradeoffs and adjudication of competing desiderata, OEBPS always takes into account actual publishing practices, business realities, and current industry dynamics. In this way it is designed to provide an actual practical solution to the problems of e-book publishing, and not just the mere possibility of such a solution.

## Terminology

The various words in common use to describe the different parts and aspects of electronic publishing can easily be seen to be much too ambiguous and vague even for general conversation—consider, for instance, the various meanings of "e-book" mentioned earlier—let alone precise enough for use in a technical specification. Consequently OEBPS, like all technical specifications, begins with a carefully defined technical terminology. These definitions are in fact a good place to begin a general overview of the specific features of OEBPS, not only because the text of the specification makes constant essential use

of these terms, but also because these terms, taken together, provide a useful initial orientation to OEBPS itself.

The following is the terminology defined in OEBPS 1.2 in the section "Definitions." In the specification itself the list is alphabetical, but we have rearranged the list here so that (with the exception of the last definition) like items are grouped together, contrasting items are adjacent, and the order has rough logic, progressing from general to particular and from data to presentation.

## OEBPS Publication

*OEBPS Definition:* A collection of OEBPS Documents, an OEBPS Package file, and other files, typically in a variety of media types, including structured text and graphics, that constitutes a cohesive unit for publication.

*Comment:* This is a fundamental concept in the specification. To summarize the OEBPS processing model: a *Publication* is processed by a *Reading System,* and the content then made available, on a *Reading Device,* to the human *Reader.* The terms Reading System, Reading Device, and Reader are explained below. OEBPS specifies conformance requirements for being a conformant OEBPS Publication, an OEBPS Document, and an OEBPS Reading System.

As described elsewhere the OEBPS Publication may be transformed into other formats before distribution to a Reading Device and so is not necessarily the format being distributed to consumers.

In so far as the OEBPS Publication is a representation of the intellectual content being published it might be considered, roughly, the "e-book," in the content sense, although if the OEBPS Publication is transformed into another format by the Reading System before being distributed to Reading Devices, then that derived format might perhaps also be reasonably called the "e-book." (We also note here that "e-book," like "book," is ambiguous between the physical book that may be bought, used, lost, etc., and the abstract edition or version of a work.)

## OEBPS Package

*OEBPS Definition:* An XML file that describes an OEBPS Publication. It identifies all other files in the Publication and provides descriptive information about them.

*Comment:* The OEBPS Package identifies and organizes the parts of an OEBPS Publication, providing the information needed for a Reading System to process the Publication. The specific contents of this important OEBPS item are discussed in a section devoted to it.

## OEBPS Document

*OEBPS Definition:* An XML document that conforms to this specification—generally containing textual content of an OEBPS Publication.

*Comment:* There are two kinds of OEBPS Documents: Basic and Extended, as defined in the next two definitions. OEBPS defines conformance requirements for conformant OEBPS Documents.

## Basic OEBPS Document

*OEBPS Definition:* An OEBPS Document that restricts itself to the markup constructs defined in this specification.

*Comment:* What this primarily amounts to is that Basic Documents use only the element names, attribute names, and attribute values of the Basic OEBPS Document Vocabulary, which is an XHTML subset. (In addition Basic Documents, like all OEBPS Documents, must meet the OEBPS Common Requirements

and the OEBPS Common Document Requirements.) Basic Documents are discussed in more detail in the section on The Component Standards, below.

### Extended OEBPS Document

*OEBPS Definition:* An OEBPS Document that uses markup constructs beyond those in this specification, but adheres to the extension mechanism defined herein.

*Comment:* An Extended OEBPS Document is an XML document that uses an XML vocabulary other than the Basic OEBPS Document Vocabulary—for instance TEI, DocBook, or an in-house XML vocabulary—or a document that mixes OEBPS Document Vocabulary with another, or several, other XML markup vocabularies. (Extended Documents, like all OEBPS Documents, must meet the OEBPS Common Requirements and the OEBPS Common Document Requirements.) Extended Documents are discussed in more detail in the section on The Component Standards, below.

### OEBPS Core Media Type

*OEBPS Definition:* A MIME media type that all Reading Systems must support.

*Comment:* OEBPS specifies six Core Media Types, two of which are defined in the specification and four of which are existing MIME Media Types. OEBPS Core Media Types are discussed in more detail in the section on The Component Standards, below.

### Reading System

*OEBPS Definition:* A combination of hardware and/or software that accepts OEBPS Publications and makes them available to readers. Great variety is possible in the architecture of Reading Systems. A Reading System may be implemented entirely on one device, or it may be split among several computers. In particular, a Reading Device that is a component of a Reading System need not directly accept OEBPS Publications, but all Reading Systems must do so. Reading Systems may include additional processing functions beyond the scope of this specification, such as compression, indexing, encryption, rights management, and distribution.

*Comment:* This concept, which is both extremely important and very easy to misunderstand, was carefully crafted to accommodate the full range of existing and possible e-book publishing strategies. A *Reading System* processes OEBPS *Publications* and presents them, on *Reading Devices,* to (human) *Readers*—but no assumptions at all are made about where or how this takes place. A Reading System can therefore include processing and transformation of content "at times prior to, or in locations distant from," its eventual presentation on the Reading Device. So in the case of a publishing process where, for instance, OEBPS is transformed by the publisher into a proprietary binary format (such as the e-book formats used by Microsoft, RCA, or Palm readers), and then that format is later distributed to Reading Devices where it can be rendered and presented for reading, the Reading System includes both the publisher's hardware and software, and the hardware and software that is the Reading Device; and it involves an intermediate transformation of content from OEBPS into a binary format. Of course Reading Systems need not be distributed in time and space, and they need not involve intermediate transformations—the entire Reading System may be contained on the Reading Device, and it may, or may not, process OEBPS Publications without transformation into an intermediate format. Reflection on likely network distribution strategies reveals that many variations are possible, and many of the differences amongst them will be invisible to the consumer. The concept of a Reading System is discussed in more detail in the section on The OEBPS Processing Model, below.

### Reading Device

*Definition:* The physical platform (hardware and software) on which publications are rendered.

*Comment:* The OEBPS specification does not specify conformance requirements for Reading Devices, which are only part of the Reading System. As described above, a Reading Device may, or may not, contain the entire Reading System. The concept of a Reading Device is discussed in more detail in the section on The OEBPS Processing Model, below.

### Content Provider

*Definition:* A publisher, author, or other information provider who provides a publication to one or more Reading Systems in the form described in this specification.

*Comment:* Although this term typically suggests publishers, it is deliberately broad and may in different circumstances refer to authors, conversion service organizations, distributors, and others.

### Reader

*Definition:* A person who reads a publication.

*Comment:* In the OEBPS specification, Reader always means a person. But be aware that elsewhere it is sometimes used to mean what OEBPS means by Reading Device or what OEBPS means by Reading System.

### Deprecated

*Definition:* A feature that is permitted, but not recommended, by this specification. Such features may be removed in future revisions.

*Comment:* This definition is here in our list of OEBPS definitions for completeness; unlike the others it pertains not to the key features of the OEBPS framework, but rather is an obligatory definition of term important to the specification itself.

## The OEBPS processing model

The basic concepts of OEBPS processing fit together like this.

1. The specific intellectual content of a book or other publication is physically represented by an OEBPS Publication (a collection of files, organized by a Package file).
2. OEBPS *Publication*s are processed by OEBPS *Reading Systems.*
3. An OEBPS Reading System includes a *Reading Device,* and the processing of an OEBPS Publication eventuates in the presentation of content, on the Reading Device, to a human *Reader.*
4. OEBPS specifies conformance standards for content (OEBPS Publications and their components), and for processing (OEBPS Reading Systems).

## The component standards

As emphasized above, OEBPS consists primarily of the coordinated applications of other standards. This section lists the most important standards applied by OEBPS and describes how they are used. In the process we will at the same time be presenting an overview of OEBPS itself.

## XML

XML is a *metalanguage,* developed and maintained by the World Wide Web Consortium (W3C), for defining descriptive markup languages. For more information on XML see the chapter on markup.

XML is fundamental to OEBPS in two ways: OEBPS is a framework for organizing and supporting the delivery and presentation of XML content; and OEBPS implements this framework with an XML DTD (the Package DTD, discussed further below). In short, OEBPS is an XML framework for XML content.

Specific applications of XML in OEBPS include:

1. All OEBPS Reading Systems must be XML processors as defined in the XML 1.0 standard.
2. All OEBPS Documents must be "well-formed" XML documents. OEBPS Documents may also be, but need not be, "valid" XML documents, with respect to an XML DTD for their particular markup vocabulary. Although XML validity is not a requirement, the authors of this chapter recommend in the strongest possible terms that content providers ensure that their OEBPS Documents are in fact valid with respect to an XML DTD. (OEBPS Documents must also conform to the OEBPS Common Requirements, and Common Document Requirements; see OEBPS section 1.4 for these requirements).
3. OEBPS Extended Documents may use any XML vocabulary, such as TEI, DocBook, ISO 12083, XHTML, or a private XML vocabulary defined by the individual content provider. However, for all XML elements not in the OEBPS Basic Document Vocabulary, at least one style rule must be provided, and at least one of the style rules provided must be from the OEBPS style language. By not allowing the formatting of XML elements (other than those in the OEBPS Basic Document Vocabulary) to take place directly, without style rule mediation, and by requiring that at least one applicable style rule be from the OEBPS style language, OEBPS ensures a high level of interoperability even for unfamiliar XML elements—without reducing the opportunity for advanced functionality and innovation.
4. OEBPS Basic Documents use the OEBPS Document Vocabulary, a subset of XHTML 1.1 (OEBPS Documents that validate against the OEBPS Document Vocabulary DTD will also validate against the XHTML 1.1 DTD). Although validity is not a requirement for conformance, we strongly recommend, again, that OEBPS Basic Documents be valid as well as well formed.
5. The OEBPS Package File is a valid XML document. (The OEBPS Package vocabulary is defined in the OEBPS Package DTD, discussed further below.)
6. W3C specifications related to XML, such as namespaces, CSS, and XHTML, are used extensively in the current OEBPS specifications and future versions of OEBPS are expected to make use of other XML-related standards (e.g., XPointer).

We emphasize again that although OEBPS does not require that OEBPS Documents be "valid" (as defined in XML 1.0) with respect to an associated XML DTD (only "well-formedness" is required by OEBPS), it is strongly recommended that publishers ensure that all their OEBPS Documents are in fact valid. Publishers who create nonvalid documents will be far more likely to eventually encounter problems with interoperability, conversion, formatting, repurposing, and use of third-party tools. These problems will be expensive to resolve and will limit opportunities for new products and features. The lesser conformance level of "well-formedness" was included in XML in order to allow for the DTD-less processing common in casual Web publishing; this lower level of conformance is entirely inadequate for high-quality financially sustainable commercial publishing.

## XHTML/Basic OEBPS Document Vocabulary

XHTML 1.1 is an XML document markup vocabulary, developed and maintained by the W3C, that corresponds roughly, in markup constructs and formatting semantics, to HTML 4.0, the SGML markup

vocabulary widely used on the Web. For more information about XHTML and HTML see the chapter on markup.

OEBPS defines a subset of XHTML 1.1 and calls this the Basic OEBPS Document Vocabulary. All OEBPS Processors are required to recognize and process this XHTML subset, and content providers may use this subset without providing style rules for its elements. This allows content providers to easily exploit current XHTML content, tools, and expertise, and also to use XHTML constructs for document components, such as tables, that might be hard to format via CSS style rules. All Reading Systems are required to process the Basic OEBPS Document Vocabulary according to the relevant portions of the XHTML 1.1 specification.

OEBPS Documents that use only this XHTML subset are called Basic Documents. An OEBPS Document that uses any markup from outside this subset is an Extended Document.

The Basic OEBPS Document Vocabulary does not include all of the XHTML 1.1 elements, attributes, and attribute values. This is partly, as with the subsetting of CSS2 (see below), in order to minimize the burden on Reading System developers and reflect hardware and software limitations. But it is also in order to align the OEBPS Basic Document Vocabulary with the current direction of development in HTML, which is away from format-oriented markup and towards content- and structure-oriented markup. For the same reason any construct deprecated in XHTML 1.1 is either deprecated or omitted from the Basic vocabulary—these are, again, typically presentation-oriented elements and in most cases the OEBPS Style Language can provide the same formatting effects via a CSS style rule. This coordination with the direction of HTML development within the W3C improves interoperability and forwards compatibility.

Although only XML "well-formedness," and not "validity," is required of OEBPS Basic documents, we, again, very strongly recommend (as does the OEBPS Specification) that publishers ensure that their OEBPS Basic Documents are not just well formed, but are also *valid* with respect to the Basic OEBPS Document DTD.

## CSS/OEBPS style language

The Cascading Style Sheets (CSS) standard, developed and maintained by the W3C, defines a mechanism for specifying the presentation of XML documents. For more information about CSS see the chapter on markup.

OEBPS defines a style language for use in OEBPS Publications that consists primarily of a large subset of constructs from CSS Level 2. Not all CSS2 properties are included in the OEBPS 1.2 style language: this is to minimize the development burden on Reading System developers and Reading Device manufacturers, who must, as described above, provide reliable performance on devices with limited resources. In addition to this CSS2 subset, the OEBPS style language also includes a few additional properties and values that provide special features that were urgently needed by book publishers—such as page layout, headers, and footers—but that were not available in CSS2.

In order to make advanced and innovative functionality possible, OEBPS Publications are allowed to use CSS2 style constructs that are not in the OEBPS style language, and may even use style constructs from other style languages. However, if they do use constructs from outside the OEBPS style language, they are required to also provide "fallback" style rules that *are* from the OEBPS style language. Allowing the use of style rules outside of the OEBPS style language supports functionality and innovation, while requiring that fallback style rules from the OEBPS style language be provided whenever style rules outside the OEBPS style language are used ensures a base level of interoperability for all OEBPS Documents—as all conformant OEBPS Reading Systems are required to support the OEBPS style language, regardless of whatever other style rules they support.

Style rules or stylesheets may be associated with an OEBPS Document in a number of different ways, with various restrictions on content and processing in order to ensure interoperability.

### XML namespaces

The XML namespace standard, developed and maintained by the W3C, defines a technique for explicitly indicating which XML vocabulary an element or attribute name is from (supporting recognition and processing) and to avoid the ambiguity that would result if a document mixed XML vocabularies that contained different elements or attributes with the same names. For more information about namespaces see the chapter on markup.

With some restrictions, OEBPS Publications may use XML namespace prefixes and the declarations that connect namespaces to the URLs that uniquely identify a namespace. OEBPS Reading Systems are allowed to process these declarations and prefixes according to the namespace specification, although they are not required to do so. However, if an OEBPS Reading System is not "namespace aware" then it is required to process a namespace "qualified name," including the colon separator that delimits the prefix and "local part," as an element or attribute name. This in effect provides partial support for the appropriate handling of namespace prefixes in Documents and DTDs even by Reading Systems that do not do namespace processing *per se.* And it is what would be expected to happen under ordinary circumstances in any case: the string of characters representing the qualified name (i.e., the concatenation of prefix, colon, and local part) is simply recognized as a element or attribute name consisting of those characters, and is processed accordingly.

To maximize interoperability: no namespace other than the XHTML namespace can be declared a "default namespace" (non-XHTML namespaces must be declared by prefix binding only); and the XHTML namespace may *not* be declared by prefix binding, but may only be declared as a default namespace. These two restrictions, taken together, ensure that OEBPS Reading Systems that are not namespace processors can still reliably recognize all XHTML elements and attributes—by their canonical XHTML 1.1 names—whenever those elements and attributes are used in a document.

In summary, although OEBPS supports and constrains the use of namespaces, there is no general requirement that namespaces be used in OEBPS Publications, even for arbitrary XML vocabularies.

### MIME media types

The MIME media typing system, developed and maintained by the IETF, is a technique for identifying content as to "media type" and a process for creating and registering new media types.

OEBPS defines a set of OEBPS Core Media Types that all conformant Reading System must support. OEBPS Publications may also include content with other media types (such as vector graphics, video, sound, scripted content, and so on), but for each resource of a media type that is not an OEBPS Core Media Type, OEBPS Publications must also include an alternative resource that *is* of an OEBPS Core Media Type. This restriction ensures that the use of innovative high-function content will not erode a baseline level of interoperability. However, the PSWG realizes that this may not be an entirely satisfactory solution for publishers wishing to produce books with multimedia or other special content. For more information on the OEBPS "fallback" strategy for media types see the section on fallbacks, below.

The OEBPS Core Media Types are given in Table 1.

## Table 1

### OEBPS Core Media Types

| MIME Media Type | Type of Data | Where Defined |
|---|---|---|
| | *MIME Media Types defined in OEBPS 1.2* | |
| **text/x-oeb1-document** | OEBPS Documents | OEBPS 1.2 |
| **text/x-oeb1-css** | OEBPS CSS-subset stylesheets | OEBPS 1.2 |
| | *MIME Media Types defined elsewhere* | |
| **image/jpeg** | raster graphics | RFC 2046 |
| **image/png** | raster graphics | RFC 2083 |
| **application/xml-dtd** | XML DTDs | RFC 3023 |
| **application/xml-external-parsed-entity** | XML "external parsed entity" | RFC 3023 |

## Unicode, UTF-8, UTF-16

The Unicode standard specifies a large encoded character set (identical with ISO 10646 implementation level 3) for almost all of the world's languages and writing systems. UTF-8 and UTF-16 are techniques for representing Unicode encoding in specific bit sequences for physical transfer (UTF-8 includes the ASCII character set encoding as a subset). Unicode is widely used in the computer industry. For more information about character sets, Unicode, glyphs, and internationalization, see the chapter on markup.

To facilitate internationalization and support multilingual documents, OEBPS publications are allowed to use the entire Unicode character set in UTF-8 or UTF-16 encodings and all OEBPS Reading Systems are required to correctly parse these encodings. However, at this time OEBPS Reading Systems are not required to provide character glyphs (appropriate visual representations) for all Unicode characters.

Version 2.0 of OEBPS will probably provide further support for glyph set negotiation between Publication and Reading System.

## Dublin Core

The Dublin Core is a simple metadata system for electronic resources. It was developed from within the library community and provides "card catalog-like" metadata to support searching and indexing. It is widely used. For more information about Dublin Core see the chapters on markup and data conversion.

OEBPS requires three Dublin Core metadata elements (Identifier, Title, and Language), but allows and supports the full set of Dublin Core 1.1 elements and provides some additional attributes specifically needed by publishers. There is, however, no assumption that the Dublin Core metadata will be adequate for all purposes, or that only Dublin Core metadata should be provided with the OEBPS Publication. OEBPS also allows the inclusion of other publication-level metadata (either conforming to other existing standards, or arbitrary systems defined by the content provider).

Version 2.0 of the OEBPS will provide a mechanism for identifying and linking arbitrary external metadata resources to an OeBF Publication, including resources that provide metadata for parts of a publication or for aspects (fonts, DRM, etc.) of a Publication. In the requirements and planning documents for OEBPS 2.0 this support is referred to as "metadata modularization."

# The package

Every OEBPS Publication contains exactly one OEBPS Package file, which specifies the OEBPS Documents, images, and other objects that make up the OEBPS Publication, along with fallbacks, metadata, and other information. The Package file, a valid XML document conforming to the OEBPS Package DTD, defines the OEBPS Publication.

## Components of the Package

Following are descriptions of each of the component parts of the OEBPS Package. When reading the descriptions below the reader is encourage to refer to the package file example that follows.

### Package identity

A unique identifier for the Publication. An attribute value for the "unique-identifier" attribute on the Package's root element (<package>) points to the Dublin Core Identifier element (required) in the metadata section whose content is then considered the primary identifier for the Publication. An optional *scheme* attribute can be used to indicate the identifier system being used, such as ISBN, DOI, etc. (NB: The value of the "unique-identifier" attribute is not the unique-identifier, but rather identifies the Dublin Core Identifier element whose content is the unique identifier.) See the identifier example in the package file example below.

### Metadata

The metadata section contains a required Dublin Core metadata record (within a <dc-metadata> element) and may also contain an optional supplemental metadata record (within an <x-metadata> record). As mentioned earlier, all Dublin Core metadata elements are supported and three Dublin Core metadata elements—Identifier, Title, Language—are required. See the metadata example below.

### Manifest

This is a list of *all* files (documents, images, stylesheets, etc.) that make up the Publication and a declaration of their media types and fallbacks, if any. The Manifest also lists files that are fallbacks for other files or objects that have unsupported media types. The Manifest supports the identification, collection, and verification of all files in a publication (to support, for instance, downloading, reading into memory, or packaging in a physical transfer format). See the manifest example below.

### Spine

The Spine is is a default linear reading order for the Publication. Because the original, and still primary, purpose of OEBPS is to support book-like publications, a linear reading order (with constraints on the media types that may occur in it) is required in order to encourage a base level of common behavior across reading systems. Only OEBPS Documents may be listed in the Spine; other media types in the publication must be referenced from them—for instance, by using XHTML elements such as <object> and <a> from a file listed in the spine, or via a chain of files that begins in a file listed in the spine. Publications may, of course, have only a single file in the Spine. See the spine example below.

### Tours

A rudimentary implementation of the "trails" functionality developed in the research hypertext systems of the 1970s and 1980s, the optional Tours are alternative reading sequences through a Publication. They may be used to support topical navigation, reader expertise levels, etc. As implemented in OEBPS 1.2, Tours have only a modest functionality; OEBPS 2.0 will have much expanded support for navigation of this sort. See the tours example below.

### Guide

The Guide is an optional list of files that contain fundamental structural components of the publication, such as table of contents, tables of plates or illustrations, foreword, bibliography, indexes, etc. Grouping these together and identifying them as to kind supports additional special processing. See the guide example below.

### Example of a Package file[Ed/CE/Comp: please preserve tabs and bold & italic in the code listing that follows]

```
<?xml version="1.0"?>
<!DOCTYPE package
   PUBLIC "+//ISBN 0–9673008–1-9//DTD OEB 1.2 Package//EN"
   http://openebook.org/dtds/oeb-1.2/oebpkg12.dtd
/>
<package
      unique-identifier="DA00042A"
      xmlns="http://openebook.org/namespaces/oeb-package/1.0/"/>
<!-- Metadata Section -->
      <metadata>
            <!-- Dublin Core Metadata Section -->
            <dc-metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
                  <dc:Title>Fish Food</dc:Title>
                  <dc:Language>en</dc:Language>
                  <dc:Identifier id="DA00042A"
                        scheme="ISBN">123456789X</dc:Identifier>
                  <dc:Creator
                        role="aut" file-as="Eisner, Ariel"
                        >Ariel Eisner
                  </dc:Creator>
                        [ Other Dublin Core elements as desired ]
                  </dc-metadata>
            <!-- User Defined Metadata Section -->
            <x-metadata>
                  <meta name="commissioning_editor" content="Bilbo" />
            </x-metadata>
      </metadata>
      <!-- Manifest -->
```

```
<manifest>
<!-- Stylesheets -- !>
    <!-- stylesheet with unsupported style constructs -- !>
        <item id="s01" href="fishbookstyles.css3"
            media-type="text/css" />
    <!-- stylesheet with supported style constructs -- !>
        <item id="s01" href="fishbookstyles.css3"
            media-type="text/x-oeb1-css" />
<!-- Major structural components -- />
    <item id="intro" href="introduction.xml"
        media-type="text/x-oeb1-document" />
    <item id="c01" href="chap1-appetizers.xml"
        media-type="text/x-oeb1-document" />
    <item id="c02" href="chap2-starters.xml"
        media-type="text/x-oeb1-document" />
    <item id="c03" href="chap3-entrees.xml"
        media-type="text/x-oeb1-document" />
    <item id="c04" href="chap3-sidedishes.xml"
        media-type="text/x-oeb1-document" />
    <item id="toc" href="contents.xml"
        media-type="text/x-oeb1-document" />
<!-- other content, including media -->
        <!-- linked from each deep fry recipe -->
            <item id="w1" href="deepfrycaution.xml"
                media-type="text/x-oeb1-document" />
        <!-- linked from frontispiece and turbot recipe -->
                <item id="ap1" href="authorpicture.png"
                    media-type="text/image/png" />
      <!-- example of short fallback chain... />
      <! -- unsupported media type; falls back to next item -->
      <item id="vv32" href="fishingclip.fli"
            media-type="video/x-fli" fallback="v19" />
      <! -- which is also unsupported, but with a fallback -->
      <item id="v19" href="fishingclip.mpeg"
            media-type="video/mpeg" fallback="p13" />
```

```
            <! -- the chain ends in an OEBPS supported media type -->
            <item id="p13" href="fishingpicture.png"
                        type="image/png" />
    </manifest>
    <!-- Spine -->
    <spine>
        <itemref idref="toc" />
        <itemref idref="intro"/>
        <itemref idref="c01" />
        <itemref idref="c02" />
        <itemref idref="c03" />
        <itemref idref="c04" />
        <itemref idref="c03" />
    <spine>
    <!-- Tours -->
    <tours>
        <tour id="tour1" title="Salmon Dishes">
            <site title="Poached Salmon"
                        href="entrees.xml#poached" />
            <site title="Salmon Cheeks in Bacon"
                        href="appetizers.xml#salmoninbacon" />
            <site title="Broiled Salmon Steaks"
                        href="entrees.xml#broiledsalmonsteaks" />
          [ and so on with more salmon recipes ]
    </tour>
        <tour id="tour2" title="Heart Healthy">
        <site title="Broiled Flounder" href ="entrees.xml#e6" />
        <site title="Poached Striped Bass" href="entrees.xml#e39" />
        <site title="Mixed Sushi" href="entrees.xml#e9" />
          [ and so on with more with more heart healthy recipes ]
    </tour>
        [ and so on with more tours of various kinds ]
    </tours>
<!-- Guide -->
    <guide>
```

```
        <reference type="toc" title="Table of Contents"
            href="toc.html" />
        <reference type="front_matter_table" title="Illustrations"
            href="toc.xml#illustrations" />
        <reference type="other.trails"
            title="Helping You Plan Your Meal"
            href="toc.xml#tours" />
    </guide>
</package>
```

## Toward OEBPS 2.0

As of this writing, the OeBF Open eBook Publication Structure Working Group is now working on OEBPS 2.0. This version, like 1.2, responds directly to the highest priority needs expressed by the publishing community, based on their experiences with OEBPS 1.0.1.

OEBPS 2.0 will provide major improvements in several areas:

### Metadata modularity

OEBPS 1.2 supports the full set of Dublin Core 1.1 metadata elements, along with other publisher-specific metadata. However, many publishers, distributors, and consumers (such as libraries, in particular) require ONIX, MARC, GILS, SCORM, or some other metadata standard that is more expressive and powerful for their particular purposes or in order to conform to the prevailing standard in a particular market or context of use. In addition, 1.2's requirement that metadata be included in the Package file is awkward and limiting in the production, distribution, and use of content. Responding to these needs, OEBPS 2.0 will provide a mechanism for allowing Publications to reference arbitrary metadata resources external to the Package file. This will allow Publications to make use of MARC, ONIX, SCORM, or any other metadata the Publisher wishes to use, to allow single metadata records to be shared by different Publications, and to allow the revision or addition of metadata without necessarily editing the Package file. Beginning with 2.0, responsibility for ongoing work in the area of OEBPS publishing metadata is now the responsibility of the OeBF Metadata and Identifiers Working Group.

This same modular mechanism will be used to provide better support for metadata in areas other than publishing, such as digital rights management, device profiling, internationalization and writing system support, and navigation.

### Internationalization

There are powerful commercial and moral reasons to ensure that electronic publishing supports all of the world's languages and writing systems. China, for instance, may initially be a bigger market for e-books than the United States. OEBPS 1.2 allows the use of the entire set of Unicode characters, requiring that all Reading Systems accept both UTF-8 and UTF-16 encodings and parse Unicode correctly, as described above. But this is still not optimal for full internationalization, primarily because the requirement to recognize those encodings does not mean the Reading System (or its Reading Devices) can necessarily render all those characters. OEBPS 2.0 will have additional support in this area, including glyph resource specification (for the appearances of characters, particularly in non-Roman character sets) and improved writing system features, including additional writing direction control to accommodate languages that read from right to left. This support will use the metadata modularization mechanisms mentioned above,

allowing improved flexibility in development, management, and sharing of internationalization resources such as glyph requirements.

### Inter- and intra-publication linking

Hypertext linking is one of the most successful and valued features of electronic reading. OEBPS 1.2 supports the use of the XHTML "<a ...>" element (in Extended as well as Basic Documents) providing basic familiar linking functionality. However, more powerful linking, such as that used in the research hypertext systems of the last 30 years, has long been requested by users; and, with the finalization of W3C XLink and XPointer standards, this advanced functionality will soon be appearing on the World Wide Web. Readers will of course expect the same functionality, or better, from their e-books. OEBPS 2.0 therefore will include substantially improved high-function linking, based on XPointer, for both links internal to a Publication, and links to other Publications.

### Navigation

Like hypertext linking, computer-supported navigation and browsing is also one of the most valued features of contemporary electronic reading. And, like hypertext linking, continued dramatic improvements in the area are expected to appear in Web browsers in the near future. OEBPS 2.0 will provide a powerful XPointer-based navigation mechanism external to the Package file, to support a variety of kinds of tables of contents, indexes, special views, and other sorts of navigation apparatus; it will also provide multilingual, graphical, and audio support as well. This system is fundamentally identical with that developed (with coordination with the PSWG) by the DAISY Consortium and released in the Talking Book standard (NISO Z39.86).

### Packaging and compatibility

Currently the PSWG is discussing revisions in the Package file to improve modularity and more effectively support ongoing backward and forward compatibility.

## In conclusion

Electronic books will happen, and they will be eventually a major part of our social and cultural lives. Given the potential advantages of e-books, current trends in technology, and current events in electronic publishing, there can be no doubt about this. However, just *when* e-books will emerge as a major commercial force is not yet clear; nor can we say for sure just what the e-book industry will be like: software, devices, content, market categories, distribution mechanisms, DRM, and the general structure of the industry are all very much still unsettled.

There are a few other things that we *can* be reasonably confident of.

One is that a flourishing industry, when it happens, will probably not be driven primarily by mass market content, but rather by content in the market categories that can make distinctive valuable use of the computational environment. This obviously means education and STM publishing, but also magazines and newspapers and certain sorts of trade content. Mass market fiction will be there, but the initial value, and the driving force behind development and adoption, will be in content that exploits the advanced features which have always been identified as providing the special value of electronic reading—and that are increasingly expected by a public for whom the World Wide Web is part of daily reading life. The nature of the future e-book is therefore best predicted not by reflecting on current e-books and imagining how they might be improved, but by reviewing the high-function experimental reading systems that have been developed in university and industry laboratories for the last thirty years, and by examining the current ambitious one-off efforts by contemporary entrepreneurs. The hard problem remains what it has

always been, not designing high-performance reading systems that deliver distinctive value to the consumer, but the next step: how to create a sustainable industry that will support this level of functionality.

Which brings us to the second thing we can be sure of. Like the publishing industry in general, the electronic book industry will almost certainly be based on SGML/XML content and production processes. Only an SGML/XML approach (or one that embodies the same fundamental principles, regardless of the specific implementation) can support both the functionality and the interoperability that are needed to create a flourishing industry. SGML/XML systems are already in wide use. The next five to ten years will probably see the nearly complete conversion of existing production processes to ones based on SGML/XML.

What about OEBPS? Will it provide the general integrating format for all e-book publishing? We think so. SGML/XML-related standards alone cannot deliver the practical level of interoperability required for financially sustainable high-performance electronic publishing. There must be further agreement on the selection and application of these standards, and there must be agreement on specific techniques for allowing innovation without sacrificing interoperability. This is what OEBPS provides. Version 1.2 of OEBPS is already very widely used in e-book publishing, and, perhaps even more important, the publishing industry appears capable of evolving new versions of OEBPS to support emerging needs and directions.

# Some advice for e-book publishers

Whether you are ready to leap immediately into the e-book fray, are considering your options, or are merely waiting until e-books become an established part of the publishing industry, you can and should act now to make your transition as smooth, efficient, and cost-effective as possible.

The first step is to learn the following.

## What you must know

- Learn about your current production workflow(s). What tools do your designers, editors, art staff, compositors, and indexers use? What electronic formats do they already produce (e.g., for delivery to the printer, or for promotional sample chapters on the Web)? Obtain and examine sample files if you can. If possible, talk to the people who actually do the work instead of their managers or supervisors.
- Learn about your current archiving procedures. What electronic files do you keep? If you outsource print production, what electronic files do your vendors return to you? Are those files binary or text? Are they revisable? Do they retain typographic quotes, structural information (such as styles or keymarking), layout, and design information? If there are last-minute corrections, are they made to the electronic files as well as the print plates?
- Learn all you can about e-book formats and production. The more you know, the better your decisions will be. Publishers draw on considerable background knowledge to make intelligent decisions about print production; making e-book-related decisions without analogous knowledge is dangerous.

## What you must do

Once you understand where you are and where you are going, you can set workflow goals:

- Wherever possible your content should be created and stored in an SGML/XML format and your workflow and publishing strategies should be organized to take advantage of the opportunities offered by SGML/XML content. You may want to adopt an existing SGML/XML element

vocabulary such as TEI or DocBook, or, if you have STM content you may want to consider a variation of one of the several widely used SGML/XML vocabularies for scientific journals. You may also prefer to develop your own SGML/XML DTD, as many publishers have. Although you can sometimes incur a small cost in lost interoperability when you invent your own SGML/XML vocabulary, as long as it is carefully designed, and, most important, consistently and correctly applied, you may not in fact lose any significant advantage—and, by having a system tuned precisely to your unique needs and opportunities you may benefit considerably in both efficiency or functionality. Although the initial investment in adopting an SGML/XML approach (whether an existing vocabulary or your own) may be greater than presentation-oriented approaches, production strategies based on SGML/XML can, all other things being equal, provide superior efficiency in process, and superior interoperability and functionality in products. That means that an SGML/XML workflow can actually produce your current products more economically and efficiently, quite apart from its benefit to future electronic products. Unfortunately, it is often the case that "all other things" are not equal and so you should...

- Respect the inherent difficulties and complexities of publishing and the accumulated wisdom of your current production process. In particular you must resist any reassuring claims that any software, or any general strategy (even SGML/XML) will easily and quickly solve your problems and immediately provide the full range of possible benefits. Although SGML/XML-based workflows have enormous advantages in the abstract, there are always difficulties in implementing XML workflows, especially in the current environment of inadequate tools and scarce general expertise. Don't be fooled by promises that this is simple. Proceed, but proceed with caution.
- Move as much of your production as possible toward a single workflow. The more consistent your workflow, the easier it is to obtain useful end products from it, and to institute consistent changes to increase its effectiveness. This does not necessarily mean investing in a gigantic end-to-end content management system, by the way; such systems are often inflexible, hindering more than they help. The goal is to extinguish unnecessary and confusing workflow variation, not to put your production methods in a straitjacket.
- Develop a house style guide for editing and composition, if you do not already have one. If you do have one, verify that it is being followed and that it does not need revision or augmentation to accommodate new directions. This guide should include keymarking abbreviations for editors and designers to use and for compositors to incorporate into stylesheets. Also, work out a scheme for canvassing production workers for needed changes to the style guide. No style guide should ever be considered so authoritative as to be beyond revision and no revision will be sustainable if it is not made in collaboration with the production team.
- Archive electronic files from your print-production process. When you move into e-books, it will cost you less to convert existing files than recreate them. Moreover, the larger your existing backlist of electronic files, the better the prices you will get from conversion vendors.
- Move your production methods away from binary, proprietary, and nonrevisable electronic formats toward text-based, open, and revisable formats. When this is not possible, try to salvage text-based and revisable files from the end of the process for archiving; for example, request .rtf files rather than (or in addition to) Microsoft Word .doc files, and QuarkXPress tags or RoustaboutXT XML files rather than (or in addition to) Quark binaries.
- Beware, however, of asking for *lossy* electronic files merely because they are text-based. Badly exported HTML, for example, may (unnecessarily) lose important information such as typesetting styles, or typographic quotes and dashes. If you are unsure of the quality of electronic files you produce or receive, examine them yourself or have someone knowledgeable do it for you.
- Outsourcing conversion or composition is often a good idea, but you *must* maintain a thorough knowledge of the production process as well as recovering all products (especially intermediate files) and insisting on complete documentation of file formats, markup vocabularies, and

conversions. Without this level of involvement and control, your costs will, eventually, soar far above what they need to be. And, in addition, opportunities for new products or innovation in functionality go unnoticed. Don't let your company be "deskilled."

- Ask for help if you need it! Find a knowledgeable consultant or trainer to explain your options and chart your progress. Hire in-house project management or production talent, or retrain existing employees. (A standards-aware Web developer should be able to move into OEBPS-based e-book production with relative ease.) Consider retaining a markup-savvy consultant to check the work your vendors are doing.
- Do not, however, offload all responsibility for developing workflows and schemas onto a consultant; what will you do when the consultant has finished the project and gone away, but no one in the company understands what was done?

# For more information

## The Open eBook Forum

The Open eBook Forum Web page is: http://www.openebook.org.

## The Open eBook Publication Structure

For more information about OEBPS see:

*The Open eBook Publication Structure—Version 1.2.* Open eBook Forum, 2002. http://www.openebook.org/oebps/index.htm.

Salo, Dorothea. *OEBPS FAQ: Introduction to OEBPS-based eBooks.* http://www.textartisan.com/oebfaq/.

## Current research and analysis: functionality

A good recent account of new directions in e-book functionality:

Schilit, Bill N., Morgan N. Price, Gene Golovchinsky, Kei Tanaka, and Catherine C. Marshall. "As We May Read: The Reading Appliance Revolution." *IEEE Computer,* 32(1), January 1999, pp. 65–73.

## Current research and analysis: users

Good accounts of recent research on e-book use:

Bell, Lori, Virgina McCoy, and Tom Peters. "E-books go to college." *Library Journal* 127 (8) (May 1, 2002): 44–46.

Bellaver, Richard F., and Jay Gillette. "The Usability of eBook Technology: Practical Issues of an Application of Electronic Textbooks In a Learning Environment." A report from Ball State University, available on-line at http://publish.bsu.edu/cics/ebook_final_result.asp.

Peters, Thomas A. "Gutterdämmerung (twilight of the gutter margins): E-books and Libraries." *Library Hi Tech,* 19(1) 2001.

Marshall, C.C. and C. Ruotolo. "Reading-in-the-Small: a study of reading on small form factor devices." *Proceedings of the Joint IEEE and ACM Conference on Digital Libraries (JCDL02),* Portland, Oregon, July 14–18, 2002.

Marshall, C.C., M.N. Price, G. Golovchinsky, and B.N. Schilit. "Designing e-books for legal research." *Proceedings of JCDL 2001* (Roanoke, VA, June 2001), ACM Press, 41–48.

## SGML/XML and the logical approach

These two articles provide an overview of the engineering model underlying the logical approach to electronic publishing, which figured prominently in the analyses of the preceding chapter. For further information about XML and XML-related standards see the chapter on markup.

Coombs, James S., Allen H. Renear, and Steven J. DeRose. "Markup Systems and the Future of Scholarly Text Processing." *Communications of the Association for Computing Machinery.* 30(11) (1987): 933–47.

DeRose, Steven J., David Durand, Elli Mylonas, and Allen H. Renear. "What is Text, Really?" *Journal of Computing in Higher Education* 1(2) (1990) pp. 3–26. Reprinted with commentary in *ACM Journal of Computer Documentation* 21(3), August, 1997

## Book-related SGML/XML vocabularies

We strongly advise new e-book publishers to consider choosing their element vocabulary from one of these two well-designed and widely used XML markup systems documented below, or to extend one of these systems.

*The TEI:* The TEI is one of the best designed and most widely used book-like XML vocabularies and there are many tools available to support its use. Although it may appear complicated at first, only the relevant portions need be used by any particular project; it is highly customizable and extensible. For simple publications, such as mass market fiction, TEI Lite, a popular TEI subset, may be appropriate. See: http://www.tei-c.org for more information. Sperberg-McQueen, C.M. and Burnard, L. (eds.) *TEI P4: Guidelines for Electronic Text Encoding and Interchange.* Text Encoding Initiative Consortium. XML Version: Oxford, Providence, Charlottesville, Bergen; 1990, 1993, 2002.

*DocBook:* DocBook is a general purpose document markup language, but particularly well-suited for technical documentation. See http://www.docbook.org/ for more information. Walsh, Norman and Leonard Muellner, *DocBook: The Definitive Guide,* O'Reilly 1999.

## Classics

For a sense of what the possibilities are, we strongly encourage reading the classics and studying the influential research systems. The early pioneering figures are Paul Otlet, Vannevar Bush, Douglas Engelbart, Ted Nelson, Andries van Dam. The advanced working systems, most of which were developed in the 1980s and many of which have functionality still not available even on the Web, include Augment, FRESS, Notecards, Intermedia, and Microcosm. An overview of the research as of 1987 is Conklin, Jeff. "Hypertext: an Introduction and Survey," *Computer* 20 (September 1987), 17–41.

These two articles inaugurated the recent phase of e-book research and development:

Kay, Alan, and Adele Goldberg. "Personal Dynamic Media" *IEEE Computer.* 10 (3), March 1977.

Yankelovich, Nicole, Noman Meyrowitz, and Andries Van Dam. "Reading and writing the electronic book." *IEEE Computer,* pp. 15–30, 18(1), October 1985.

## Wider issues

Lynch, C. "The Battle to Define the Future of the Book in the Digital World." *First Monday: A Peer-Reviewed Journal on the Internet* 6, 6 (June 2001).
http://www.firstmonday.dk/issues/issue6_6/lynch/index.html.

Although the authors have been involved in the development of the Open eBook Publication Structure, they write this article as individuals, and therefore the views presented here should not be considered in any way official with

respect to the OeBF. In particular, no part of this chapter should be understood as authoritative with respect to the OEBPS specification itself. The text of that specification is the only authoritative expression of the intention of its authors and the OeBF. Finally, the authors emphasize that this presentation of the OEBPS is intended only as a general orientation to the OEBPS specification; in the interests of clarity, brevity, and value to the intended audience, it is not only incomplete but in several places presents a simplified version of the technical aspects of the specification.