DHQ: Digital Humanities Quarterly
2010
Volume 4 Number 2

Twisty Little Passages Almost All Alike: Applying the FRBR Model to a Classic Computer Game

Jerome McDonough <jmcdonou_at_illinois_dot_edu>, Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign
Matthew Kirschenbaum <mgk_at_umd_dot_edu>, Maryland Institute for Technology in the Humanities, University of Maryland
Doug Reside <dreside_at_umd_dot_edu>, Maryland Institute for Technology in the Humanities, University of Maryland
Neil Fraistat <fraistat_at_umd_dot_edu>, Maryland Institute for Technology in the Humanities, University of Maryland
Dennis Jerz <jerz_at_setonhill_dot_edu>, English — New Media Journalism, Seton Hill University

Abstract

Humanities scholars and librarians both confront questions regarding the boundaries of texts and the relationships between various editions, translations and adaptations. The Functional Requirements for Bibliographic Records (FRBR) Final Report from the International Federation of Library Associations has provided the library community with a model for addressing these questions in the bibliographic systems they create. The Preserving Virtual Worlds project has been investigating FRBR's potential as a model for the description of computer games and interactive fiction. While FRBR provides an attractive theoretical model, the complexity of computer games as works makes its application to such software creations problematic in practice.

Introduction

Humanities scholars have continually confronted questions regarding the boundaries of the texts that they study and the complex inter-relationships that can exist among various editions, printings, translations, and adaptations — in short, the versions — of a work. While librarians have long recognized the distinction between a work as an intellectual creation and its embodiment within a particular physical form (and the need to adequately describe both), the publication of the Functional Requirements for Bibliographic Records Final Report by the IFLA Study Group on the Functional Requirements for Bibliographic Records (FRBR) marked a pronounced increase in the level of attention that the library community has devoted to these issues. FRBR proposed a formal model for bibliographic description that recognizes four classes of entities as implicated in descriptive practice: Works (unique intellectual or artistic creations), Expressions (the realization of Works), Manifestations (the physical embodiment of particular Expressions), and Items (single exemplars of a Manifestation). Attributes commonly found in bibliographic description, such as publisher or title, are bound in the FRBR model to one of these four entities. 1

In the decade since the Final Report was issued, a tremendous amount of discussion has occurred regarding the interpretation of FRBR and its appropriate application within bibliographic systems. At the same time, there has been almost no cross-communication between humanities scholars engaged in the kind of work described above (“textual studies,” as it is called) and library specialists. In fact, discussions of distinctions between various ontological states occupied by a textual object well predate the genteel deliberations of textual scholars and IFLA study groups alike. “[W]hen composition begins,” wrote [Shelley 1903, 39], “inspiration is already on the decline, and the most glorious poetry that has ever been communicated to the world is probably a feeble shadow of the original conceptions of the poet.” These conceptions have typically been recast as the author’s “intentions” by 20th-century editors seeking to adjudicate between different versions of a poem or novel by appealing to their ability to intuit what the author would have wished, could he or she only be given the opportunity to declare it once and for all. The so-called eclectic editions that resulted — standardized texts that were in fact composites of any number of multiple, surviving documentary instances of the work — also gave rise to an elaborate philosophical framework which is perhaps most clearly articulated by [Tanselle 1989] in his tripartite distinction among works, texts, and documents. The vocabulary here is striking: “Photocopying a manuscript book or a printed book creates a new document, the latest in the series of attempted reproductions of the work its text represents...” [Tanselle 1989, 54] Tanselle’s discourse on works, texts, and the documents in which they are manifest clearly would not seem out of place in the FRBR report; his view of textuality, which reinforced and extended the writings of key Anglo-American bibliographers such as W. W. Greg and Fredson Bowers, was only seriously challenged in the closing decades of the 2

20th century, when editors such as D. F. McKenzie and Jerome McGann advanced theories that laid stress on the interaction between individual textual artifacts and the larger social and material fields in which they are embedded. One is finally interested not in a definitive text but in the documentary text.

Within the world of traditional manuscripts and print publications, the relationships between the various versions of a particular text are already extraordinarily complicated; applying these existing categories to new forms of creative electronic texts (including interactive fiction and computer games) makes these relationships become even more vexing and difficult to describe than we had anticipated. Because each individual or subsequent encounter with the same interactive work can generate different outputs, the adequacy of traditional descriptive models applied by librarians to enable scholars' access to textual materials needs to be carefully examined. From the scholar's (or teacher's) perspective, even mundane activities such as a textual citation or assigning students a particular passage to read become problematic. Moreover, even the simplest electronic “text” is in fact a composite of many different symbolic layers, from microscopic traces on physical storage media up through machine code, higher-level languages, and finally the visible characters one actually reads on a screen. Of course without the kind of preservation that comes from recognition of such creations as part of our late 20th-century cultural heritage, any such issues will be rendered moot for future generations since the work will not survive in any accessible or recoverable form. 3

The Rochester Institute of Technology, Stanford University, the University of Illinois at Urbana-Champaign and the University of Maryland are currently investigating the preservation of computers games and interactive fiction. Sponsored under the Library of Congress' National Digital Information Infrastructure for Preservation Program (NDIIPP), this project seeks to identify the specific difficulties in the preservation of computer games and interactive fiction that distinguish them from other forms of digital information we wish to preserve, to develop metadata and packaging practices to allow us to manage the long-term preservation of these digital materials in a manner consistent with the Open Archival Information System Reference Model, and to test those practices via ingest of computer games and interactive fiction into a set of functioning digital repositories. 4

The project employs a case set methodology, focusing on a limited number of computer games and works of interactive fiction chosen to highlight a variety of potentially problematic issues. The works were intentionally chosen to represent a variety of different periods in computer history, different computing platforms, different styles of artistic work, and different intellectual property issues. Works within our case set include 5

- *Spacewar!* — The first graphical computer game, *Spacewar!* is a space combat simulation based on E. E. “Doc” Smith's *Lensman* series of books, created in 1962 at MIT for the PDP-1 computer. It was later used as the basis for two different commercial arcade video games and has been ported to a number of different platforms, including the Atari 2600 and more recently the iPhone.
- *ADVENTURE* — One of the most influential interactive fiction works, ADVENTURE was created by Will Crowther at BBN Planet in 1976 and expanded upon by Don Woods at Stanford. The availability of the game's source code on the early Arpanet has led to its being modified by a number of individuals to add new puzzles, traps and monsters, and to it being ported to innumberable different programming languages and operating systems.[1]
- *Star Raiders* — Originally published by Atari for the Atari 400 and 800 computers, a modified version of *Star Raiders* would become one of the most popular games for the Atari 2600 game console and be ported to the Atari 5200 and Atari ST machines, with a relatively unsuccessful sequel, *Star Raiders II*, released in 1986.
- *Mystery House* — Created by Roberta and Ken Williams in 1980 for the company which would later be named Sierra Online, this is the first interactive fiction work to employ computer graphics. A binary executable version of the original game for the Apple II system was put into the public domain in the late 1980s, but the apparent loss of any source code for the game limited development of derivative versions, until the *Mystery House Taken Over* project produced and released a reverse engineered version of the game in 2005.
- *Mindwheel* — Published by Brøderbund Software in 1984, this work by Poet Laureate Robert Pinsky is also notable for being a compound analog/digital work, containing both a print novella and the game software.
- *Doom* — Published by iD Software in 1993, this game came to define the first-person shooter style of game, revolutionized 3D graphic display technology for games, and as a result of both the game's design and iD Software's decision to open source the original game engine led to a whole new Internet culture of game customization and modification. Like ADVENTURE, Doom has been ported to a large number of different operating systems, including OS X, all versions of Windows, Linux and Android, as well as various console platforms including the Atari Jaguar, Game Boy Advance, Nintendo 64, Sega Saturn, Sony Playstation and Nintendo SNES systems.
- *Second Life* — Launched in 2003 by Linden Lab, this has become one of the most popular of the non-role playing game multiuser virtual worlds. Our project is focusing on the preservation of several islands contained within *Second Life*, including the International Spaceflight Museum and Stanford Humanities Lab's Hotgates Island. Given that

> islands in Second Life exhibit on-going development and change and can be the work of a number of different individuals, preserving any island in Second Life is more akin to trying to preserve a collaborative performance art piece while it is being produced than trying to preserve a data file.

The first phase of our project, which we have recently completed, has examined the games in this case set to try to identify representative issues they present for the long-term preservation of computer games and interactive fiction. As a particular test of existing library practices we have been examining the application of the FRBR entity-relationship model to computer games and interactive fiction, including the seminal work ADVENTURE. FRBR, a model developed primarily to assist in end-user access to library materials may seem an unusual choice for a project concerned with digital preservation, but as [Thibodeau 2002] has trenchantly stated, “In order to preserve a digital object we must be able to identify and retrieve all of its digital components” [Thibodeau 2002, 12]. A fundamental aspect of any effort to preserve digital resources is thus the development of systems to describe and track the components of a digital work and to relate works (and their physical embodiments) to each other, including describing the provenance of manifestations as a work evolves over time. Such activity is important not only for librarians and archival caretakers, but also (as we have seen) for scholars, including those who may in the future wish to produce the equivalent of a critical edition for a computer game, as well as a wide variety of more casual users, including hobbyists, fans, and enthusiasts. This paper will examine the difficulties encountered by the project in seeking to apply the FRBR entity relationship model within the realm of computer games, and our project’s suggestions for “pretty good” practices for the application of FRBR and traditional bibliographic descriptive practices to this ever-evolving electronic genre. 6

Functional Requirements for Bibliographic Records & Their Discontents

FRBR is, at first glance, a promising mechanism for representing this twisty little maze of cultural heritage. It is an entity-relationship model capable of discriminating among changes to the substance or “content” of the work, as well as its physical embodiment in particular carrier media. In a traditional FRBR representation, one might start with the work that is *Hamlet*. The different versions of the play that are extant are the work's expressions. These expressions are realized in manifestations, i.e. the folios and quartos that have survived, as well as the more modern editions based upon those sources. A discrete artifact that one holds in hand, for example the copy of the Arden Shakespeare sitting nearby on my bookshelf, is an item. FRBR also recognizes the possibility of more complex relationships between the various types of entities it enumerates. A copy of Plays and Poems of Shakespeare [Shakespeare 1878] may constitute a single item exemplifying a single manifestation, but that manifestation embodies multiple expressions and works. The various FRBR entities may also recursively contain one another; the rock musical version of Hamlet by Czech musician Janek Ledecký [Ledecký 2000] contains a number of individual songs, each of which can be considered as works in their own right. 7

The quartet of Work, Expression, Manifestation and Item form what the FRBR report calls the Group 1 entities. Group 2 entities are “Person” and “Corporate Body,” the entities which create Works, realize Expressions, produce Manifestations, and own Items. Group 3 entities define different types of subject matter with which a Work may be concerned, and include “Concept,” “Object,” “Event” and “Place.” The FRBR report also notes that Group 1 and Group 2 entities may serve as the subject matter of Works as well. 8

In addition to defining the basic relationships between Group 1 entities discussed above, the FRBR report also describes a number of other possible relationships that may exist between Group 1 entities. Table 1 shows the various possible relationships between the Group 1 entities enumerated in the FRBR report. A number of these relationships can be of use when describing video games and interactive fiction. If we consider a game franchise such as the Doom series from iD software, for example, we can easily find examples of successor relationships between Works (the original Doom is succeded by Doom II), supplemental relationships (the original Doom is supplemented by the Doom Wiki (http://doom.wikia.com), and even transformation relationships (the original game Doom was transformed into the movie version starring Dwayne “The Rock” Johnson). Other Group 1 relationships are also easy to identify in the case of Doom. The original shareware expression of Doom has a revision relationship to the full, registered commercial expression (with the full registered version containing two weapons, the plasma gun and the BFG9000, not available in the shareware version). An expression-to-expression translation relationship also exists between the source code implementation of the original Doom game engine and a binary executable version of the game engine compiled from that source code for a Windows '95 machine. The DVD manifestation of the movie Doom has an alternate manifestation, the Blu-ray manifestation. The item consisting of our library's copy of Doom 3 for the PC platform has two other items as parts: a CD-ROM containing the software, and a print manual. 9

	Work to Work	**Expression to Expression**	**Expression to Work**	**Manifestation to Manifestation**	**Manifestation to Item**	**Item to Item**
has a successor	X	X	X			
has a supplement	X	X	X			
has a complement	X	X	X			
has a summary	X	X	X			
has adaptation	X	X	X			
has a transformation	X	X	X			
has an imitation	X	X	X			
has part	X	X		X		X
has an abridgement		X				
has a revision		X				
has a translation		X				
has an arrangement (music)		X				
has a reproduction				X	X	X
has an alternate				X		
has a reconfiguration						X

Table 1. Table 1 - Relationships between FRBR Group 1 Entities

10 At the same time that FRBR seems to promise a useful and detailed modeled for description of library materials, including games, certain long-standing challenges still exist even with more traditional applications of the FRBR model. For example, there is no formal consensus on how much of the work has to change before a new expression is declared. Catalogers (for FRBR is primarily a cataloger's tool) are asked to rely upon common sense, community practice, and other heuristics. Catalogers are all too aware, however, that even textual materials within libraries can raise complex issues with respect to the question of “how different” a particular text must be to qualify as a new edition.

11 In the case of an electronic object, the complications proliferate almost exponentially [Renear 2006]. At first it might seem that all versions of ADVENTURE should be grouped under a single “Work,” a particular instance of the game (the last version modified by Don Woods, for instance) should be the “Expression,” a particular file with a unique MD5 hash should be the “Manifestation,” and an individual copy of that file (perhaps on a Commodore 64 664 Block disk) would be the “Item.” But what if the text read by the reader is exactly the same, but the underlying code is different? These variants might be simple (a comment added to the FORTRAN source code), peripheral (such as the ability to recognize “x” as a synonym for the command “examine”), or very large (a port of the code from FORTRAN to BASIC). Should these code level variants be considered different expressions? To further complicate matters, what if the FORTRAN code were exactly the same but compiled to two different chips? For example, an IBM mainframe and a Commodore 64 might both have a FORTRAN compiler, but the two compilers will interpret the FORTRAN to a different set of machine instructions. It might also be the case that two FORTRAN compilers designed by different programmers will generate slightly different machine language. Even the same compiler might generate slightly different machine code from a single source code file depending on the options with which it is invoked. Should these compiled executables, different in their binary structure but based on the same FORTRAN code, represent different “Manifestations” or different “Expressions”?

12 Finally, even two files with exactly the same MD5 signature participate in a larger software environment at runtime. The drivers that run the display interface, the keyboard, the memory, and the disk drives arguably become part of ADVENTURE when the user is playing the game. For instance, the experience of playing the game using the 6507 chip in a Commodore 64 hooked up to a black and white television may be different than the experience of playing the game on the same chip in a Commodore SX64 (the all-in-one machine some felt fit to call “portable”). Should the software environment on which the binary is executed be a part

of the classification scheme at all? Would playing the game on a video monitor (which displays only a fixed number of lines at a time) provide a substantially different experience from a session with the same game played on a Teletype (which saves the output indefinitely on paper)?

We have applied the FRBR model to several different and specific instances of ADVENTURE: the source and data files for the original Don Woods version, as well as two early variants produced by Will Crowther, retrieved on April 27, 2008 at 6:01 pm from Dennis Jerz's server (http://jerz.setonhill.edu/if/crowther/), as well as the DOS Windows executable of these files edited to compile under GNU g77, a free FORTRAN compiler (http://www.russotto.net/%7Erussotto/ADVENT/). This work will be presented in the course of the paper, together with rationale and discussion in the context of the kind of issues enumerated above. We will also discuss the significance of this work for the broader digital humanities community, insofar as it represents the intersection of library and information science, textual studies, and software forensics. 13

As more and more libraries and repositories begin the process of collecting born-digital objects, they will invariably encounter material that transcends the boundaries of documents, email, and other more or less conventional forms of electronic records. ADVENTURE, as both a working computer program and as a virtual world, as well as an artifact with widespread popular interest, is a harbinger of the kind of content which increasingly needs to be accessioned, cataloged, and described. FRBR represents the library community's best effort to date to distinguish between different versions and editions of a work. We believe the work discussed represents an important test case for FRBR's applicability to complex born-digital objects. 14

ADVENTURE's Passages

Will Crowther originally developed the game ADVENTURE in the mid-1970s while he was working at BBN, the company responsible for launching the ARPANET. The game focuses on the exploration of a cave complex in which a variety of puzzles and hostile antagonists (including an axe-throwing dwarf) must be defeated. Crowther made a compiled version of the game available through his BBN account, and a copy ended up in the hands of Don Woods, a graduate student at Stanford University. Woods contacted Crowther and obtained from him a copy of the game's FORTRAN source code, which he modified to change the game play, adding several additional fantasy elements. The game, as modified by Woods, was widely distributed on the ARPANET and was a significant influence on early hacker culture, with phrases from the game such as “a maze of twisty little passages, all alike” and the magic word “xyzzy” having been appropriated and re-used in a variety of contexts. The game provided the first instance of a new genre of work, interactive fiction. It also sparked the creation of a slew of successors as other programmers picked up the source code distributed by Woods and modified it to suit their taste (whether in terms of game play or programming language of choice).[2] 15

The game's wide distribution and immense popularity in the early days of networked computing, along with the ready availability of the FORTRAN source code modified by Woods, led to a proliferation of new versions of the game as programmers ported it to new languages, new operating platforms, and modified its structure to add new puzzles, monsters and territory. Figure 1 shows a very partial family tree for ADVENTURE[3], starting with the original Crowther and Woods versions and showing the path of succession as particular versions of the software are picked up by other programmers and modified. Of particular importance in this tree is the variation in types of descent. We can identify three major types of change that can occur when a programmer takes a pre-existing version of ADVENTURE and modifies it. One of the more common forms of variation occurs when the programmer ports the source code from one programming language into another (or into a significant variant of the original language), while making no changes (or as few and as minor as possible within the scope of changing the programming language) to the game play itself. You can see many instances of this in Figure 1, such as the transition from the original Don Woods FORTRAN IV version from 1977 to the Jim Gillogly port to the C language in 1993. Another type of change is one in which the programming language is maintained from one version to the next, but the source code is modified to change the game play. The transition from the original Will Crowther version of ADVENTURE to Don Wood's version would qualify as such a modification. While the programming language was still FORTRAN, Woods modified the game to add new antagonists and puzzles. The final type of change is the most extreme, a reimplementation in which both the source code language and the game play are modified. An example of this would be the reimplementation of ADVENTURE that Don Woods undertook using the C programming language in 1995.[4] 16

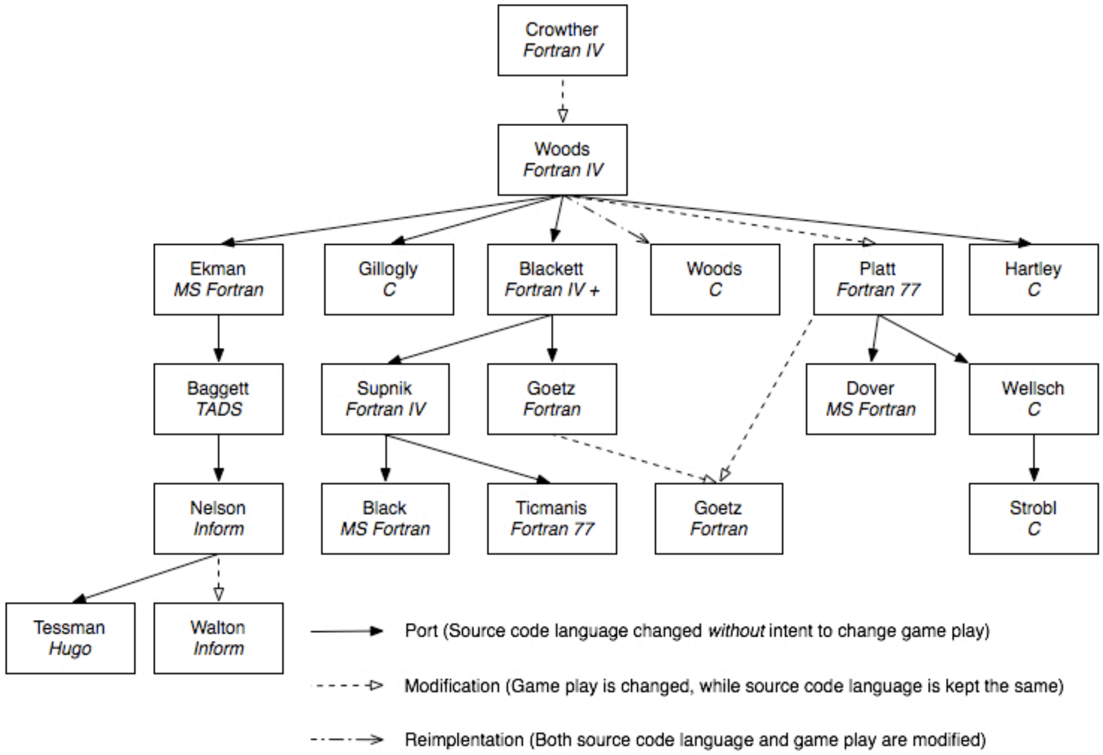


Figure 1. A partial family tree of ADVENTURE, showing ports, modifications and reimplementations.

This typology of changes highlights one of the unique features about interactive fiction works such as ADVENTURE as textual artifacts, and one of the difficulties they present for those trying to describe them within the bounds of the FRBR model. A port of a game from one language to another involves a significant amount of creative, intellectual effort and results in the creation of a source code “text” which, while implementing similar algorithms, may otherwise bear very little resemblance to the original source code on which it is based. Significant variations in the source code, however, may result in no visible changes in the game play presented by a compiled executable of the new code. If we consider the transition from the original Don Woods FORTRAN version to the Gillogly C language version as programmers looking at the source code, we see enough changes to qualify the Gillogly version as not merely a new expression in the FRBR sense, but in all probability an entirely new work. The Gillogly version certainly seems to answer to the FRBR criteria that when “the modification of a work involves a significant degree of independent intellectual or artistic effort, the result is viewed, for the purposes of this study, as a new work” [IFLA 1997, 18]. From the point of view of a player interacting with the game, however, the two different versions are practically identical. So, in determining whether something constitutes a new work or expression in the world of interactive fiction, should we assume the point of view of the programmer, or the point of view of the game player? 17

While it is tempting to try to resolve this question through reference to user needs, different users will have very different needs when approaching gaming materials, and those differing needs will have a profound impact on the users' preferred intellectual organization for games. From the point of view of someone interested in playing the game, an executable prepared from the Gillogly source code and one prepared from the Woods code are essentially equivalent and of equal interest. From the point of view of a programmer interested in game programming techniques in the C language, the two could not be more different. From the point of view of a scholar of game history, the two are different, but highly inter-related. Establishing the dividing lines between Group 1 entities in FRBR has always been a somewhat subjective matter, but interactive fiction (and perhaps software generally) highlight the way in which differing and incompatible subjectivities may reside in a library's or archive's patrons. 18

A closer examination of some of the specific instances of the game ADVENTURE reveals further complexities for those seeking to apply the FRBR entity-relationship model to the description of computer games. In our research, the earliest version of ADVENTURE that we have been able to examine is the original FORTRAN version created by Will Crowther, consisting of a FORTRAN source code file which is 727 lines in length, and a separate 733 line data file containing a dictionary of terms that the game employs to interpret user commands, a set of textual responses provided by the game in response to user commands, and the geometry of the virtual world which the player explores. While the FORTRAN source code file does contain a few comments, these serve only to describe the operation of the code; there is nothing resembling bibliographic metadata in either file, with no 19 authorship or date of creation provided. The files in question were retrieved by Dennis Jerz with Don Woods' cooperation from a backup tape of Don Woods' student account at Stanford University, and are named advf4.77-03-11 and advdat.77-03-11. Research by [Jerz 2007] indicates that the date contained within the filenames is probably an indication of when Don Woods obtained the source code from Will Crowther. The backup tape also contained two new versions of the FORTRAN source code created by Don Woods, modifications of the code provided by Will Crowther. These two new source code files were named advf4.77-03-23 and advf4.77-03-31. There was also a new version of the game data file that was created by Woods, named advdat.77-03-31. The changes made to the source code by Woods in the two later files were relatively minor and do not reflect the more significant changes that he made in the version he eventually distributed. The changes between the advf4.77-03-11 version and the advf4.77-03-23 version consisted of changing one line of existing code and adding another 16 lines of new code. The new lines of code appear to implement an external FORTRAN function that Will Crowther's code had invoked, but which was not available on the PDP-10 system that Woods was using. Table 2 shows the differences between the two versions, with modified code in the March 23rd version italicized, and new code in bold face.

advf4.77-03-11	*advf4.77-03-23*
CALL SHIFT(A(J+1),7*(I-6),YY)	CALL SHIFT(A(J+1),7*(K-6),YY) **SUBROUTINE SHIFT (VAL,DIST,RES) IMPLICIT INTEGER (A-Z) RES=VAL IF(DIST)10,20,30 10 IDIST=-DIST DO 11 I=1,IDIST J = 0 IF (RES.LT.0) J="200000000000 11 RES = ((RES.AND."377777777777)/2) + J 20 RETURN 30 DO 31 I=1,DIST j = 0 IF ((RES.AND."200000000000).NE.0) J="400000000000 31 RES = (RES.AND."177777777777)*2 + J RETURN END**

Table 2. Table 2 - Differences between W. Crowther's and D. Wood's FORTRAN IV source code

The differences between Don Wood's version, dated March 23, 1977, and the one dated March 31, 1977, were of about the same magnitude, with 14 lines of code in the March 23rd version modified in the March 31st version, and one line in the March 23rd version deleted. The differences between the original game data file, advdat.77-03-11, and the modified version by Woods, advdat.77-03-31, have no impact on game play whatsoever and consist solely of changes in numeric identifiers assigned to twelve terms contained in the game's dictionary. 20

The files retrieved from Don Wood's old student account can be viewed as comprising three different versions of the game ADVENTURE. The first is the original Will Crowther version consisting of the two files advf4.77-03-11 and advdat.77-03-11. The second is a version with the FORTRAN source code file advf4.77-03-23 modified by Don Woods but with the original Will Crowther data file of advdat.77-03-11. The third is a further modification by Woods consisting of his FORTRAN source code file named advf4.77-03-31 and his modified data file advdat.77-03-31. However, while there are modifications to source code or data file (or both) for all three of these versions, they are all essentially identical in terms of game play. While Don Woods would later further modify ADVENTURE to add new monsters and puzzles, these early changes do not include those later changes to the game. A player engaged with executable programs prepared from these three source code versions would insist that they are, in fact, all the same. 21

Given that all three versions present an identical “text” to the game player, and substantially similar text to an individual reading the source code, an analysis of these three instances might conclude that all represent the same FRBR work. However, the issue of FRBR expression is somewhat more complicated. Certainly there are changes in the source code and data files between the three versions, and given their relative historic importance within game studies, recognizing them as distinct expressions seems reasonable. However, each of the versions, if compiled and played, would be indistinguishable from an end user perspective. Or 22 at the very least, the game play would be indistinguishable. From the perspective of a user interested in experiencing the actual original game, these many instances are effectively a single expression.

Our discussion so far has somewhat glossed over the fact that our hypothetical game-playing end user would not be interacting with a source code file, but an executable file created from a FORTRAN source code file using a compiler. FRBR states that “Inasmuch as the form of expression is an inherent characteristic of the expression, any change in form (e.g., from alpha-numeric notation to spoken word) results in a new expression. Similarly, changes in the intellectual conventions or instruments that are employed to express a work (e.g., translation from one language to another) result in the production of a new expression” [IFLA 1997, 20]. A compiler translates human readable text into machine code; it therefore alters both the language of the work and the notation employed. Any executable expression a user can actually interact with is clearly a distinct expression from the source code used to create it, albeit one that can be produced algorithmically from the source code expression. 23

This gives us at least some basis for considering the three versions of the game as unique and separate expressions, even when viewed from the game player's perspective. The executables compiled from the three different versions of source code will not be identical. They will contain minor variations in their structure and file size that will be visible to an end user should they choose to investigate it. On that basis, an argument could be made that each version of the game constitutes a unique expression, even from the point of view of the game player, and that if we combine the five original files in our possession with compiled executables of the source code files, we have six expressions of a single work, as seen in Figure 2. But at some level, this does not seem a very satisfactory result. Users interested in game play will be more likely to consider the actual interactive experience provided by the software to constitute the basis for determining whether something is a unique expression, not its size in bytes or the internal structure of the op codes contained within an executable file. 24

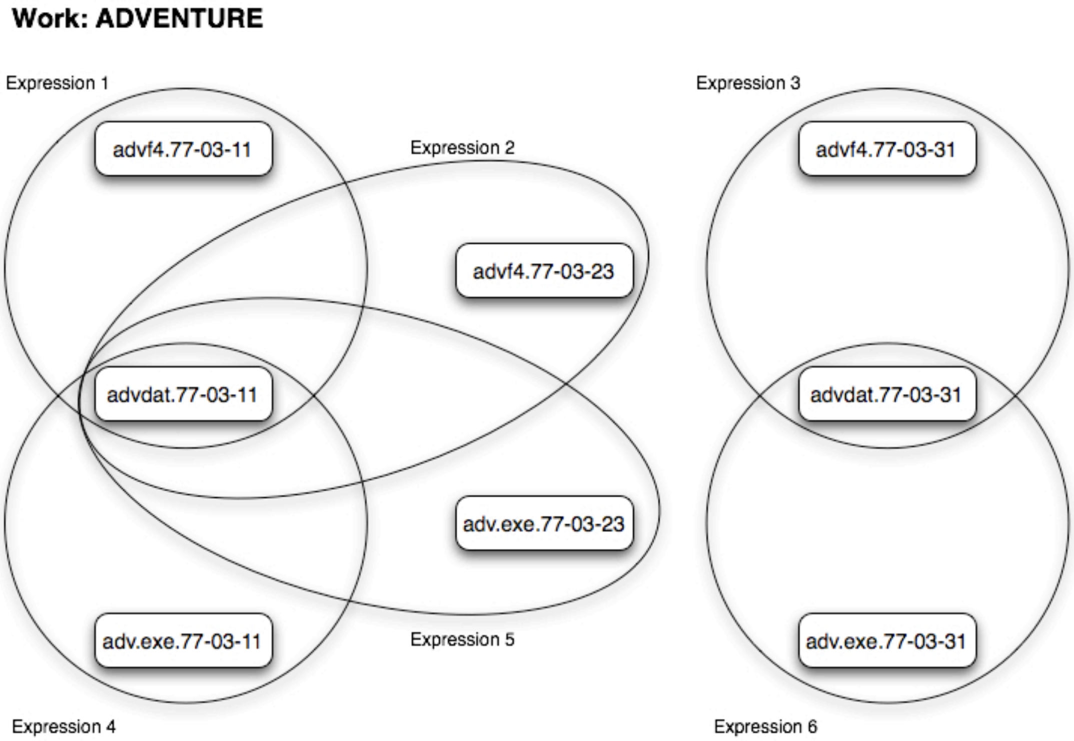


Figure 2. FRBR Expressions of ADVENTURE

Unfortunately, an alternative in which we claim that all three source code expressions produce a single executable expression with three different manifestations is difficult to model accurately in existing bibliographic systems. It also runs afoul of the relationships established between Group 1 entities in the FRBR model. Figure 3 demonstrates the problem. It is possible to establish a set of Group 1 entities for the versions of ADVENTURE in our possession that includes a single FRBR Expression for the executable versions of the program, but in order to express the relationship between the source code versions and their executable derivatives we need to state that a particular source code manifestation has a translated form in a particular executable manifestation (as noted by the dashed arrows). While this may be a relatively accurate assertion, the FRBR model as expressed in [IFLA 1997] only provides for asserting a translation relationship at the level of FRBR Expressions, not at the Manifestation level. Thus, the only model for the different versions of ADVENTURE we have in hand appears to be one in which we have six 25

Expression-level entities for ADVENTURE (3 source code Expressions, and 3 corresponding executable Expressions), a model that may impede the search efforts of end users interested in playing the game.

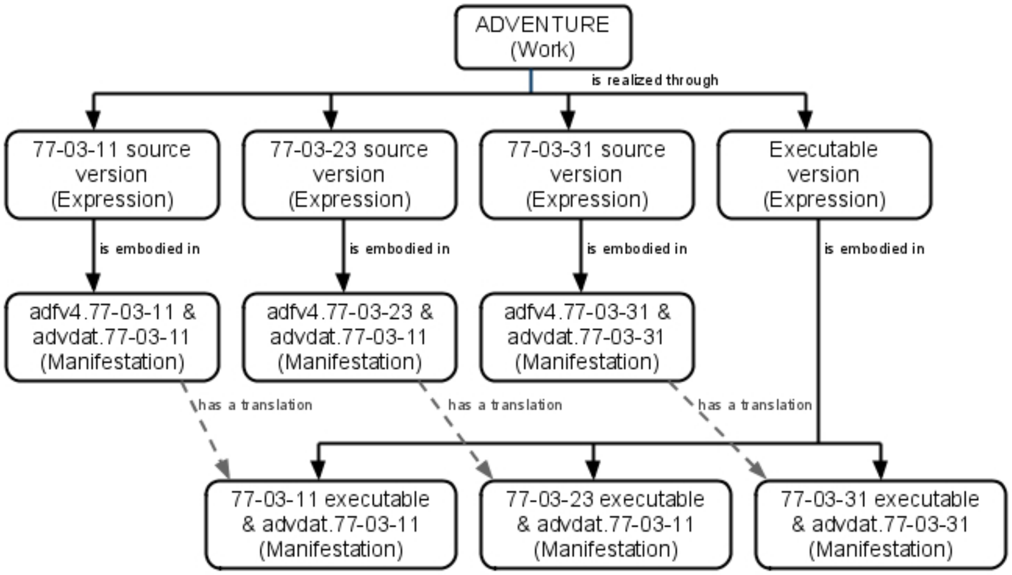

ADVENTURE
(Work)
is realized through
77-03-11 source version (Expression)
77-03-23 source version (Expression)
77-03-31 source version (Expression)
Executable version (Expression)
is embodied in
is embodied in
is embodied in
is embodied in
adfv4.77-03-11 & advdat.77-03-11 (Manifestation)
adfv4.77-03-23 & advdat.77-03-11 (Manifestation)
adfv4.77-03-31 & advdat.77-03-31 (Manifestation)
has a translation
has a translation
has a translation
77-03-11 executable & advdat.77-03-11 (Manifestation)
77-03-23 executable & advdat.77-03-11 (Manifestation)
77-03-31 executable & advdat.77-03-31 (Manifestation)

Figure 3. ADVENTURE with a Single Executable Expression

26 The compilation of the three different source files here raises other questions with respect to the nature of games and their description within a FRBR framework. Crowther wrote the original ADVENTURE FORTRAN source code file to compile on a DEC PDP-10 running the TOPS operating system. It includes a call to an external function, IFILE, used to read in the data file (which was to be named “TEXT”). The IFILE function was not part of the FORTRAN language used on the TOPS-10 and TOPS-20 operating systems [Digital Equipment Corp. 1987], and can be found at statement 1001 in the source code:

```
1001 LTEXT(I)=0 I=1 CALL IFILE(1,'TEXT')
```

27 As the source code defines a variety of other subroutines within itself, we can assume that IFILE was probably part of an external library of FORTRAN functions in use on the PDP-10 at the time the game was written.[5] This example is similar to the case mentioned previously, where Don Woods reimplemented a function named SHIFT that apparently was an extrinsic function called by Will Crowther's code. These cases demonstrate that the boundaries of what constitutes the software are not equivalent to the boundaries of the files that a programmer might identify as constituting the source code for the game. Even in this relatively early game, we see programmers beginning to rely on libraries of functions distributed with compilers and operating systems to simplify the job of authoring code. In modern computer games, programmers are even more reliant on libraries provided by third parties to simplify the development of game software. Compiling code into an operational copy of the game requires not just the code developed by the game author, but also all of the code present in any third party library functions that might be invoked by the game. Warcraft III, for example, relies on DirectX libraries distributed by Microsoft for various multimedia functions.

28 Computer games do not possess the clear boundaries of a physical artifact such as a book. Games (and all software) are embedded in and intertwined with a technological environment that includes compilers, linkers, code libraries, operating system facilities, and various kinds of hardware. A functioning copy of a computer game requires not only the game software but also a complete computer system. When we set out to describe a game within the FRBR framework, we immediately confront the question of “what constitutes the game,” and that question can be difficult to answer. The case of ADVENTURE reveals that some of the complexity involved in answering that question is due to the fact that the game is a compound work containing a variety of

subsidiary FRBR works, authored by different people at different times for different purposes. The IFILE library function presumably was not written to be part of ADVENTURE; it was written to provide file I/O services for any program. But it formed an essential component of the game, which could not be compiled (or played) without it.

The FRBR framework allows for these types of whole/part relationships at both the Work and Expression levels, and modeling computer games using these types of FRBR relationships actually allows us to assert a variety of relationships between different entities that might be of interest to end users (see the transformation and translation relationships in Figure 4). However, practical application of FRBR in the world of computer games could easily prove to be a tremendous burden on those describing games. A complete description of a computer game within the FRBR framework would need to identify all of the various subsidiary Works constituting the games' technological components, whether created by the game author or not, delineate the relationships between all of the different components, and provide some level of intellectual description for each. For a game like ADVENTURE, defined by one source code file and one data file (at least if we limit ourselves to a single instance), identifying implicated components such as compile-time libraries might be slightly onerous but on its face does not appear an impossible task. But in more modern games, containing thousands of files created by dozens or hundreds of individuals working at a variety of different companies and distributed as parts of different products, complete description within a FRBR framework would be an insurmountable burden on current cataloging resources. 29

Unfortunately, there are several credible use scenarios for description that require this fine level of detail. For those concerned with preservation of computer games, the need for some level of description down to the individual computer file level is a real concern. Librarians and archivists concerned with copyright issues may need to be aware that the putative creator of a game may not be the only creative agent involved in production of game software and that subcomponents of a game may have differing intellectual property status. Scholars studying games may be quite interested in patterns of use and reuse of game components among both game companies and game players, and without fine-grained description, investigating these issues will become much more difficult. 30

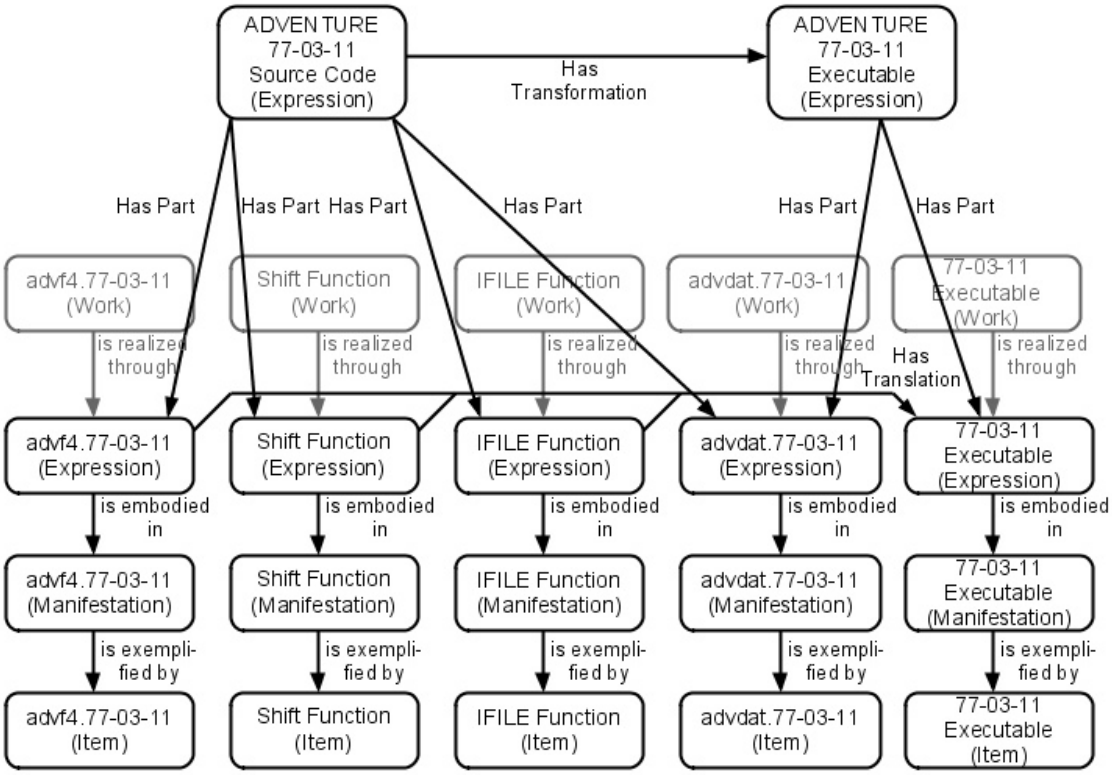
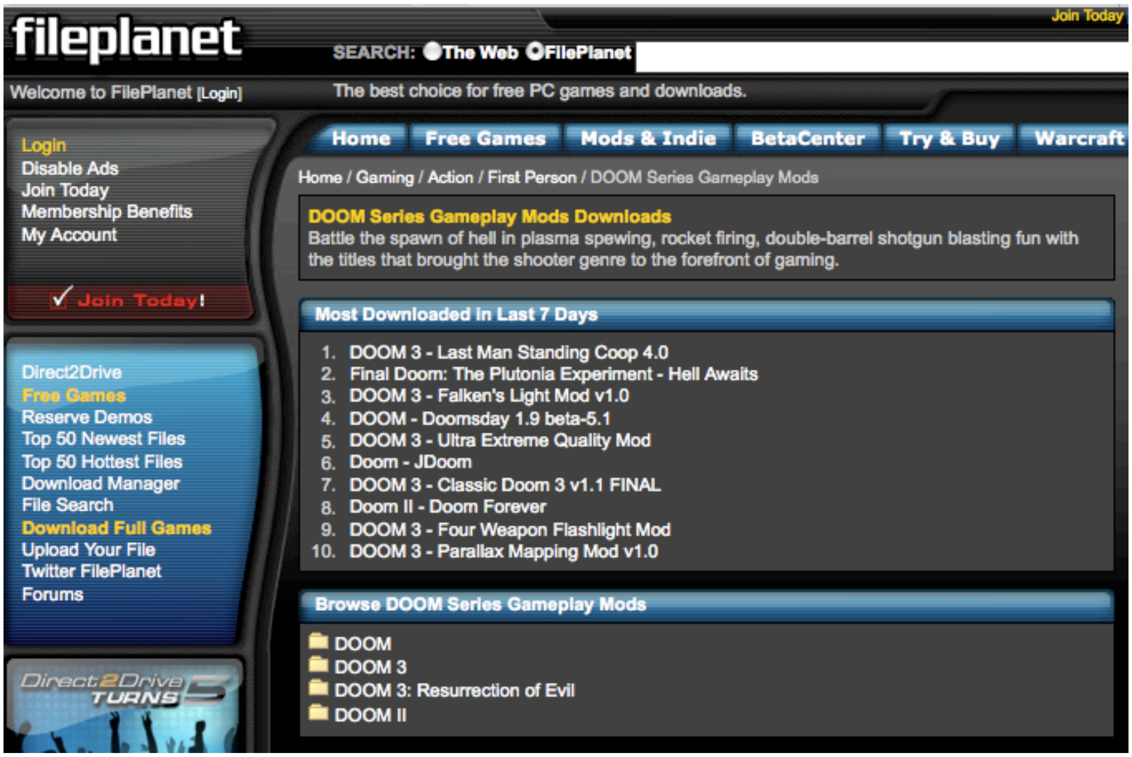

ADVENTURE 77-03-11 Source Code (Expression)
Has Transformation
ADVENTURE 77-03-11 Executable (Expression)
Has Part
Has Part
Has Part
Has Part
Has Part
Has Part
advf4.77-03-11 (Work)
Shift Function (Work)
IFILE Function (Work)
advdat.77-03-11 (Work)
77-03-11 Executable (Work)
is realized through
is realized through
is realized through
is realized through
is realized through
Has Translation
advf4.77-03-11 (Expression)
Shift Function (Expression)
IFILE Function (Expression)
advdat.77-03-11 (Expression)
77-03-11 Executable (Expression)
is embodied in
advf4.77-03-11 (Manifestation)
Shift Function (Manifestation)
IFILE Function (Manifestation)
advdat.77-03-11 (Manifestation)
77-03-11 Executable (Manifestation)
is exempli-fied by
advf4.77-03-11 (Item)
Shift Function (Item)
IFILE Function (Item)
advdat.77-03-11 (Item)
77-03-11 Executable (Item)

Figure 4. FRBR Group 1 Entity Relationships for two ADVENTURE Works

The description of computer games within the FRBR model provides a reasonably compelling justification for the notion of a Superwork, a potential addition to the set of FRBR entities that would collocate multiple Works under a single descriptive banner for the purposes of retrieval. As Figure 1 makes clear, for a game like ADVENTURE, the number of instances of a game which 31 can be said to be, in Svenonius's terms, "similar by virtue of emanating from the ur-work" [Svenonius 2000, 38] can be very large and continue to grow over a period of decades after a game's initial release. While a legitimate case can be made that many of the versions of ADVENTURE could be placed under the banner of a single Work, our research on game preservation has also led us to examine games such as Doom from id Software where a culture of “modding” [6] has arisen among the user community, and in such cases the number of separate works, all emanating from the original Doom ur-Work (and in many cases employing the original game engine), can easily number in the hundreds. The ability to collocate these resources is important for users: they care quite a bit about which version of Doom was used in generating a particular mod, and they also wish to be able to distinguish Doom mods from mods of other games such as Quake. An examination of gaming web sites such as FilePlanet (see Figure 5 [7]) show that the user community for games already engages in collocation activities for themselves that group games not only by a game series but within a series by a particular edition. A variety of Doom mods have been written to work with specific source ports of the original Doom engine, so the ability to collocate mods that work with a specific implementation of Doom is important to game players. Given the users' obvious desire to collocate works associated with a particular engine, with a particular version of Doom, and across the entire Doom series, application of a Superwork entity may be the simplest means of enabling users' preferred mode of searching.



Home / Gaming / Action / First Person / DOOM Series Gameplay Mods

DOOM Series Gameplay Mods Downloads

Battle the spawn of hell in plasma spewing, rocket firing, double-barrel shotgun blasting fun with the titles that brought the shooter genre to the forefront of gaming.

Most Downloaded in Last 7 Days

1. DOOM 3 - Last Man Standing Coop 4.0
2. Final Doom: The Plutonia Experiment - Hell Awaits
3. DOOM 3 - Falken's Light Mod v1.0
4. DOOM - Doomsday 1.9 beta-5.1
5. DOOM 3 - Ultra Extreme Quality Mod
6. Doom - JDoom
7. DOOM 3 - Classic Doom 3 v1.1 FINAL
8. Doom II - Doom Forever
9. DOOM 3 - Four Weapon Flashlight Mod
10. DOOM 3 - Parallax Mapping Mod v1.0

Browse DOOM Series Gameplay Mods

- DOOM
- DOOM 3
- DOOM 3: Resurrection of Evil
- DOOM II

Figure 5. Doom Series Game Mods Page from fileplanet.com.

32 One final comment should be made about the FRBR model and its application to computer games and interactive fiction. While the focus of our analysis has concentrated on description of games within the framework of the FRBR Group 1 entities, the fact that we are working on a project involving software preservation has meant that we have had to devote a certain amount of attention to intellectual property issues. It is unfortunate, in our view, that the FRBR model does not mention intellectual property rights in discussing the relationships that exist between Group 1 and Group 2 entities (person, corporate body). Given the examples set forth in the IFLA Study Group's report, e.g.,

> **W1** Franz Schubert's Trout quintet
> *e1* the composer's notated music
> *e2* the musical work as performed by Rosina Lhevinne, piano, Stuart Sankey, double bass, and members of the Juilliard String Quartet
> [IFLA 1997, 20]

it would appear that copyright can adhere at the level of both the Work and the Expression. If not, it would be difficult to account for cases such as a recorded song that possesses both a recording performance production copyright and a creator copyright within the FRBR framework. Greater clarity on how the IFLA Study Group conceived of intellectual property rights fitting into their model would be of great benefit to those trying to work on games. While ADVENTURE is not a particularly problematic game with respect to this issue, having passed into the public domain, modern games can have extremely complicated rights situations, in which music with separate performance and creator copyrights are included in a game copyrighted by yet another entity.[8]

Conclusion

Our research has found that the FRBR model provides a mechanism capable of describing some of the web of relationships that exist between games and between the component parts that make up a game. This is impressive given the sheer extent of the component parts and the intricacy of their relationships in any modern computer game or interactive fiction work. However, there are a variety of both practical and theoretical problems that must be addressed when seeking to apply FRBR in the world of computer games and interactive fiction. While the practical issues may be susceptible to technological solutions, the theoretical ones will require further development of the FRBR standard if it is to realize its promise as a descriptive mechanism for these types of interactive art. 33

The relationships between Group 1 entities in the FRBR model, when applied to computer games such as DOOM, tend to favor descriptions composed of a number of different Works, rather than a number of different Expressions under the banner of a single Work. This may make it more difficult for searchers to collocate variants of a game under a single banner. While asserting relationships between Works (particularly successor relationships) may alleviate this problem somewhat, identifying all of the relationships to record may be difficult and time-consuming. Game aficionados may be willing to invest the time and energy in deciphering relationships between instances of a game such as those shown in Figure 1, but asking catalogers to engage in this level of detail may be unrealistic. 34

The multiplication of Work-level entities in the description of games and interactive fiction is further promoted by the need to describe the various component pieces of games individually. This in turn leads to a need to assert even more whole/part relationships between Expression-level entities. Each of these constituent parts can obviously come with their own set of additional Work-to-Work relationship issues, primarily to indicate succession relationships. 35

All of the preceding makes practical application of the FRBR model to computer games a time-consuming and expensive enterprise. This is not necessarily a fault of FRBR; the reality is that detailed description of games, particularly within a preservation environment, is time-consuming and expensive. Our work has been carried out within a context of ensuring the preservation of games in the long-term, and that requires fairly detailed description of the software, including identification of its component parts and dependencies on other software and hardware. FRBR provides a theoretical framework that can be applied to this task, but it cannot in itself lessen the costs associated with such detailed description. 36

Our work with ADVENTURE and other computer games, however, does highlight two deficiencies in the FRBR entity-relationship model. The first problem arises from the complex tangle of derivative works associated with any particular game. As [Smiraglia 1992] noted, “Bibliographic families can be as complex as human, genealogical families. Many generations can exist on the same plane at the same time” [Smiraglia 1992, 72]. Computer games and interactive fiction provide ample evidence of this, with the number of derivative works created for some games easily numbering in the hundreds. Neither catalogs as they exist today nor FRBR provides sufficient facilities to ease collocation of these works for users. Computer games provide one of the stronger arguments for the concept of a Superwork and adding support for Superworks to our bibliographic systems, and to the FRBR model. The second problem is the omission of any mention of intellectual property rights within the FRBR model. While [IFLA 1997] made it clear that they were not enumerating every attribute of or relationship existing between bibliographic entities, the failure to account for intellectual property relationships between Group 1 and Group 2 entities is extremely problematic for those attempting to describe computer games, and we suspect much other digital material. While on its face, copyright might be seen to constitute a relationship between a FRBR Expression entity and a Group 2 entity, case law on copyright has wavered over the years with respect to how it handles the distinction between uncopyrightable ideas and copyrightable expressions [Samuels 1989], particularly with respect to software. Alignment of legal theory and cataloging theory regarding the separation between artistic/intellectual creations and their expression in particular forms is, we suspect, a difficult task that will require the input of the both communities. 37

ADVENTURE and its passages also offer a compelling demonstration of the extent to which complex born-digital objects, especially those that are popular, historically significant, or cherished by communities of enthusiasts, will demand other kinds of expertise not likely to reside within a typical cataloging department. Our work applying FRBR to ADVENTURE required an advanced knowledge of antiquarian computers, systems, and programming languages, as well as an appreciation for how the game has been ported and reworked by diverse constituencies over the course of several decades. Digital humanities is well 38 suited to serve as a disciplinary rubric for uniting these disparate kinds of interests and expertise, and we believe that the bibliography of complex electronic objects must become an increasingly significant aspect of activity for those who consider themselves its practitioners.

Notes

[1] Versions of ADVENTURE have already been released for both the Android operating system (see http://www.ecsoftwareconsulting.com/node/11) and the iPhone/iPad (http://itunes.apple.com/us/app/advent/id284446752?mt=8).

[2] See [Jerz 2007] and [Montfort 2003] for detailed discussions of ADVENTURE's origins and significance.

[3] The information in this figure is based on a listing of ADVENTURE variants compiled by [Dalenberg 2006]. Note that some of the instances of ports shown here may involve migration between different versions of a programming language. So, the transition from Blackett's version of ADVENTURE to Supnik's involves migrating from two variant implementations of the FORTRAN IV programming language.

[4] It should be noted that the difference between what is considered a port, a modification or a reimplementation is a matter of both degree and interpretation. The transition from Wellsch's C language interpretation to Strobl's was a matter of moving from an implementation intended to work on Unix-style operating systems to one intended to work on Microsoft Windows machines. While Strobl did not intend to alter the game play, he did put a Windows graphical user interface on top of the game that allowed the user to respond to various requests for game input via buttons. Whether this constitutes a major change in game play depends on which aspects of the original you consider significant and which you do not.

[5] We have not been able to identify a FORTRAN library for the TENEX and TOPS-20 operating systems in use at BBN at the time Will Crowther authored ADVENTURE that contained an IFILE function. However, there was an IOFIL FORTRAN library created for the DECSystem-20 that contained an IFILE function to support software written for the DEC F40 FORTRAN IV compiler for the PDP-10. A report listing software converted from DECSystem-10 to DECSystem-20 mentioning the library can be found at http://pdp-10.trailing-edge.com/decuslib20-01/01/decus/20-0000/conversion-status.mem.

[6] "Modding" refers to taking an existing game and making modifications to it to alter game play in some fashion. The game Doom, like ADVENTURE, kept the data which was displayed to the user in a separate file. In the case of DOOM, this data file (known as a WAD file) was reverse engineered by the gamer community, which then started modifying it to add new monsters, game levels, weapons and other changes.

[7] Image taken from http://www.fileplanet.com/31773/90004/section/DOOM-Series on Oct. 7, 2009.

[8] The language of FRBR with regards to Item/Group 2 entity relationships is also somewhat problematic for those dealing with game collections. While "owned by" is described within the IFLA Study Group report as including either ownership or custody of an item, it is highly unusual to find a computer game (or other software) which a purchaser will actually own. They instead obtain a license to possess and use the software. The use of "owned by" as a relationship term could be seen as misleading to users if presented in the context of software collections.

Works Cited

Dalenberg 2006 Dalenberg, Russell. *Versions and Ports of Adventure known to exist. v.2.6.7 Russel Dalenberg's Home Page*. 2006. http://www.io.com/~ged/www/advelist.html.

Digital Equipment Corp. 1987 Digital Equipment Corp. *TOPS--10/TOPS--20 FORTRAN Language Manual. AA-N3838-TK, AD-N3838-T1*. Maynard, MA: Digital Equipment Corp., February 1987. http://bitsavers.org/pdf/dec/pdp10/TOPS10_softwareNotebooks/vol11/AA-N383B-TK_fortLangMan.pdf.

IFLA 1997 IFLA Study Group on the Functional Requirements for Bibliographic Records. *Functional Requirements for Bibliographic Records Final Report*. The Hague, Netherlands: International Federation of Library Associations and Institutions., Sept. 1997, as amended through Feb. 2009. http://www.ifla.org/files/cataloguing/frbr/frbr_2008.pdf.

Jerz 2007 Jerz, Dennis G. "Somewhere Nearby is Colossal Cave: Examining Will Crowther's Original "Adventure" in Code and in Kentucky". *Digital Humanities Quarterly* 1: 2 (2007).

Ledecký 2000 Ledecky, Janek. *Hamlet*. Prague: BMG Ariola ČR, 2000.

Montfort 2003 Montfort, Nick. *Twisty Little Passages: An Approach to Interactive Fiction*. Cambridge: MIT Press, 2003.

Powers 2000 Powers, Richard. *Plowing the Dark*. New York: Farrar, Strauss, and Giroux, 2000.

Renear 2006 Renear, Allen. "Is An XML Document a FRBR Manifestation or a FRBR Expression? - Both, Because FRBR Entities are note Types, But Roles". Presented at *Extreme Markup Languages 2006* (August 7-11, 2006). *Proceedings of Extreme Markup Languages 2006, Montreal Quebec, August 7-11, 2006* (2006). http://idealliance.org/papers/extreme/Proceedings/html/2006/Renear01/EML2006Renear01.html#tod0e5.

Samuels 1989 Samuels, Edward. "The Idea-Expression Dichotomy in Copyright Law". *Tennessee Law Review* 56 (1989), pp. 321-462.

Shakespeare 1878 Shakespeare, William. *The Plays and Poems of Shakespeare; with One Hundred and Seventy Illustrations; from Designs by Eminent Artists*. Edited by A.J. Valpy. London: Bell & Daldy, 1878.

Shelley 1903 Shelley, Percy Bysshe. *A Defense of Poetry*. Edited by Albert S. Cook. Boston: Ginn & Company., 1903. http://books.google.com/books?id=u2s3qZXXsIAC&dq=inauthor%3A%22Percy%20Bysshe%20Shelley%22%20%22when%20composition%20begins%22&client=firefox-a&pg=PR1#v=onepage&q=inauthor:%22Percy%20Bysshe%20Shelley%22%20%22when%20composition%20begins%22&f=false.

Smiraglia 1992 Smiraglia, Richard Paul. *Authority Control and the Extent of Derivative Bibliographic Relationships*. Chicago: Graduate Library School, University of Chicago, 1992.

Svenonius 2000 Svenonius, Elaine. *The Intellectual Foundation of Information Organization*. Cambridge: MIT Press, 2000.

Tanselle 1989 Thomas, Tanselle G. *A Rationale of Textual Criticism*. Philadelphia: University of Pennsylvania Press, 1989.

Thibodeau 2002 Thibodeau, Kenneth T. “Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years”. In *The State of Digital Preservation: An International Perspective*. Washington DC: Council on Library and Information Resources, 2002. http://www.clir.org/pubs/reports/pub107/thibodeau.html.