

IFS – The Interactive Fiction Schema

I. Critique

When I decided to create a metadata scheme for interactive fiction, I did so with the intention for my scheme to be used for a large IF website in a manner similar to Baf's Guide to the IF Archive. After reviewing my classes and properties, I think that my scheme could function in this capacity given some modification.

One issue I ran into when constructing this schema was the problem of whether to borrow properties from other schemas or to create my own sub-properties. For a while I had an `ifs:title` property, but since there were no further restrictions on it compared with `dcterms:title`, I removed that property from my schema and replaced it with `dcterms:title` where it had been used in my records. I decided that it would be best for the schema to borrow as much as possible, and only define new properties when there was a specific purpose. This would be a benefit for two reasons: first, it minimizes the amount of work that I had to do for the creation of the schema and minimizes the cost of future maintenance and upgrades; second, it allows for a greater amount of interoperability since the scheme borrowed from was DC Terms. This means that an application does not have to know how to interpret IFS in order to get some value from the records.

I am a little torn over this as it means that records using this schema will have to reference the `dcterms` schema as well, leaving another place for error. The extra namespace also increases the cognitive burden on the cataloger ("Was the publisher element native to IFS or DCTerms? I can't remember.") and increases the chance of data input error. Still, my hope would be that data entry would be handled by a script connected to a graphical user interface that would prevent such errors by automatically writing out the records from form fields.

I also struggled with naming conventions for the various properties and classes. For a while I had the `ifs:hasMedia` property to describe specific resources associated with the Work, as well as `ifs:hasGraphics` and `ifs:hasSound` to describe the general presence of sound or graphics and

explain their necessity using the Requirement class. Given that `hasMedia` describes a completely different concept than `hasGraphics`, I felt that the later was poorly named and renamed it to `graphicalComponent`. Even then I am not completely satisfied with some of the names that I chose to represent concepts.

When examining the use of properties, I found that many properties were not used very often. For instance, the `ifs:theme` and `ifs:editor` properties were not used at all. Other elements were used only for either Physical or Digital resources (e.g., `ifs:downloadLink`, `ifs:browserLink`, and `ifs:documentationLink` were never used on the Physical resources). Some of the properties were not used because they were irrelevant; or the information was unavailable, while others were not used because the content information was not easily accessible (see below).

Another issue was not with the schema or its implementation, but rather with the generation of the records: works of interactive fiction are by their very nature closed and difficult to catalog. A cataloger can pick up an atlas and flip through the book, seeing that it covers North America, South America, and Australia; but cataloging the content for a work of interactive fiction must progress through the possibilities to discover all that a work has to offer. This is an unrealistic way to catalog these works. The restrictive nature of the format severely limits the depth of content description. I spent between half an hour and an hour on each record, and I feel like I barely scratched the surface of the content itself. Because the IFS has a large number of content-related properties I felt like I could not take full advantage of the schema without pouring hours upon hours into each record. Authors may be able to provide some of the information (e.g., `ifs:theme`, or `ifs:challenge`), but authors may also be hesitant to give too much information about the surprises within their works.

There were a few properties that I would add to the schema. For instance, I realized while creating records for my ten items that I have no property to represent whether or not a work has mature content which would be useful given the variance in the maturity levels of IF. But again, this would be difficult to catalog without direct help from the authors themselves or a solid corps of community members.

Finally, finding the appropriate information was also more of a task than I thought it would be. I searched Wikipedia, IFWiki, the IF Archive, Baf's Guide, as well as publisher or author

pages to find the most complete records I could. However, this also gave rise the problem of managing versions: I could not always tell which version a particular site was describing. This was something that I had not thought about when I first envisioned the schema. I thought that it would be best to manage the intellectual concept of the specific work, rather than a specific version. While it is possible to list additional dates or contributors or release versions, there is no way to align a contributor to a specific version of the work without creating an entirely new record.

Overall, I think that the schema has potential to be useful for IF. However, the greatest hurdle does not lie with the schema itself, but rather with its implementation.

II. List of Resources to Describe

1. The Google Adventure
⌚ <http://blogoscoped.com/googleadventure/>
2. Blue Lacuna
⌚ <http://www.lacunastory.com/>
3. Anchorhead
⌚ <http://wurb.com/if/game/17>
⌚ http://en.wikipedia.org/wiki/Anchorhead_%28video_game%29
4. Adventure
⌚ <http://www.rickadams.org/adventure/>
⌚ http://en.wikipedia.org/wiki/Colossal_Cave_Adventure
5. Zork I: The Great Underground Empire
⌚ <http://www.infocom-if.org/games/zork1/zork1.html>
⌚ http://en.wikipedia.org/wiki/Zork_I
6. Zork II: The Wizard of Frobozz
⌚ <http://www.infocom-if.org/games/zork2/zork2.html>
7. The Hitchhiker's Guide to the Galaxy
⌚ <http://www.infocom-if.org/games/hhgtag/hhgtag.html>
8. Ad Verbum
⌚ <http://nickm.com/if/adverbium.html>
9. The Cave of Time
⌚ http://en.wikipedia.org/wiki/The_Cave_of_Time
10. Shadow Unit
⌚ <http://shadowunit.org/gettingstarted.html>

III. List of Properties and Classes

A. DC Terms properties utilized in IFS:

- ⌚ identifier
- ⌚ title
- ⌚ publisher
- ⌚ rights
- ⌚ description

B. Native IFS properties (indentation denotes sub-property hierarchy) :

- ⌚ author
- ⌚ composer
- ⌚ artist
- ⌚ editor
- ⌚ date – expects W3CDTF
 - i. dateFirstPublished
 - ii. dateModified
 - iii. dateLastModified
- ⌚ challenge
- ⌚ genre
- ⌚ theme
- ⌚ review
- ⌚ competition
- ⌚ award
- ⌚ choiceMethod
- ⌚ forgivenessLevel
- ⌚ participation
- ⌚ language – expects RFC4646
- ⌚ hasMedia
- ⌚ coverArt
- ⌚ soundComponent
- ⌚ graphicalComponent
- ⌚ release
- ⌚ technology
 - i. originalTechnology
 - ii. authoringPlatform
 - iii. availableOnPlatform
 - iv. requiresTechnology
- ⌚ producedAs
- ⌚ browserLink
- ⌚ downloadLink
- ⌚ documentationLink
- ⌚ size
- ⌚ relatedWork

C. Classes native to IFS

- ⌚ Work (subclass of the DCMI Type InteractiveResource)
- ⌚ Media
 - i. Visual
 - ii. Audio
- ⌚ Technology

D. Controlled Vocabularies (Implemented as Class structures)

- ⌚ ChoiceMethod
 - i. Branching
 - ii. TextParserTwoWord
 - iii. TextParserFullSentence
 - iv. Other
- ⌚ Requirement
 - i. Required
 - ii. Optional
 - iii. None
- ⌚ Challenge
 - i. Puzzle
 - ⌚ WordGames
 - ii. Maze
 - iii. Combat
- ⌚ ProducedAs
 - i. Digital
 - ii. Physical

