MMET: A Migration Metadata Extraction Tool for Long-term Preservation Systems

Feng Luan, Mads Nygård

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Trondheim, Norway
Email: {luan, mads}@idi.ntnu.no

Abstract. Migration is the most often used preservation approach in long-term preservation systems. To design a migration plan, custodians need to know about technical infrastructure about a preservation system, characteristics and provenance about digital materials, restrictions about preservation activities, and policies about retention rules. However, current tools cannot provide all these information. They just can output information about formats and characteristics for several given formats. Hence, in this paper, we design a migration metadata extraction tool. This tool uses the stored metadata to retrieve the above information for the custodians. The test results show that due to the limitation on the stored metadata, our solution still cannot get the sufficient information. However, it outputs more migration metadata and has better performance than current tools.

Keywords: Migration, Metadata Extraction, Long-Term Preservation System

1 Introduction

Our society is becoming an e-society where computing technology is indispensable in people's life. For example, government departments use e-government systems to create digital government documents, education institutions use e-learning systems to provide and digitize teaching resources, and libraries use e-library systems to store and publish digitized books, magazines, pictures, etc. When so much information is digitized or is born in a digital form, preservation becomes an important issue for the information management science.

Several preservation approaches, such as migration, emulation, universal virtual computer, encapsulation, and technique preservation, have been proposed and analyzed in [1-7]. Amongst them, migration is the most often used approach. Also it is deemed as the most promising preservation approach. In practice, when doing a migration, custodians of a preservation system must prepare a plan, test the plan, and deploy the plan. For example, [8,9] introduce methodologies to detect format obsolescence, and [10,11] introduce methodologies to select migration solutions. In these methodologies, one of prerequisites is to obtain necessary and sufficient

information about technical infrastructure about the preservation system, characteristics and provenance about every type of digital materials, restrictions about preservation activities, and policies about retention. Hence, we choose our research question on how to get this information.

We find that several tools have been designed for this purpose. They can scan a file system and extract metadata from files in the file system. However, these tools have some drawbacks. For instance, 1) they take much time to do the extraction; 2) the extracted metadata may not be accurate; and 3) the custodians just get format information and characteristics for several given formats. In order to overcome these drawbacks, we design a new migration metadata extraction tool (MMET), which uses a set of administrative metadata to obtain necessary information for migration.

The structure of this paper is summarized as follows: We firstly in Section 2 introduce several related works and our research motivation. Secondly, our previous works on migration data requirements are shown in Section 3. The requirements would be used in MMET to specify what metadata should be retrieved. Thirdly, we summarize the design of MMET in Section 4. Fourthly, we test MMET and evaluate it with JHOVE in Section 5. Finally, a further discussion about our solution and current solutions is shown in Section 6.

2 Related Work and Motivation

Current solutions to retrieve migration metadata are based on digital materials. There are three sets of such solutions. The first set focuses on the extraction of content characteristics. For example, the eXtensible Characterization Language (XCL) [12] can extract characteristics of a digital material, and can further use XCL-ontology to compare the characteristics before and after migration; ExifTool [13] can read, write and modify metadata that are embedded with the digital materials; Tika [14] can extract metadata and structured text content from various types of digital materials.

The second set is to extract metadata about format. The format extension may be a clue for judging a format. However, since the file extension is modifiable, this kind of the judgment may not be trustworthy. The custodians have to use other mechanisms. For instance, in Linux files are assigned a unique identifier of a given format, so that the FILE command can use this identifier to judge a format rather than the file extension. The second example is DROID [15] that use internal and external signatures to identify the format of a digital material. These signatures are stored in a file downloaded from the format register PRONOM [16]. Using the DROID signature, the custodians can query a given format in PRONOM, and then can view the technical context for this format. Fido [17] converts the signature downloaded from PRONOM into regular expression for obtaining a good performance on the format identification task.

The last set combines the functions of the above two classes. JHOVE [18] is such example. It is designed to identify a format, validate a format, extract format metadata, and audit a preservation system. JHOVE is able to support 12 formats, e.g., AIFF, ASCII, BYTESTREAM, GIF, HTML, JPEG, JPEG-2000, PDF, TIFF, UTF-8,

WAVE, and XML. In addition, JHOVE provides an interface by which developers can design modules for other formats. Some projects have integrated JHOVE into their solutions. For example, AIHT [19] is a preservation assessment project. In their assessment procedure, JHOVE is used to identify formats in a preservation system and calculate the number of files for each format. PreScan [20] is another implementation of JHOVE. Using PreScan, preservation metadata can be automatically and manually created and maintained. Another example of the last set is FITS (Format Information Tool Set) [21]. It contains a variety of third-party open source tools, such as ExifTool, JHOVE, DROID, and the FILE command.

The above solutions output metadata mainly about format and characteristics. Obviously, it is not sufficient in terms of our previous research work on migration data requirements (see Section 3). Lacking sufficient metadata, it might cause some problems when designing a migration procedure. For instance, 1) the migration procedure may fail, as digital materials are encrypted; and 2) the custodians over estimate migration time, so that they may choose a fast but expensive solution. In addition, the extraction of characteristics is time-consuming. As mentioned in [20], it takes PreScan about 10 hours to extract characteristics metadata from 100 thousand files.

Hence, we try to find a solution that should be more efficient and get more metadata than current solutions. When surveying preservation systems, we find that the most systems have stored many metadata together with digital materials. These metadata provide description information, structural information, and administrative information. Hence, we decide to use these stored metadata to retrieve necessary information for migration. In the following sections, we will summarize the design of this tool.

3 Quality Data Requirements on Migration

The preserved metadata may use various types of metadata schemas. In order to help the custodians identify what metadata element is necessary to be extracted, we first designed 24 migration data requirements (see Figure 1) in our previous work [22]. Secondly, we did a survey to validate the necessity and sufficiency of the requirements. The details about the requirements and the survey comments are summarized as following:

- **Storage**: Storage metadata provide background about components in the storage system, such as its storage medium (R1), its storage driver (R2), and its storage software (R7)[1]. Using those metadata, the custodians can find compatible storage solutions or replacements. There are two conditions under which metadata of this category are necessary: 1) Preserved digital materials are offline data. The related storage system may be seldom accessed, so that people in future may not know the components of the storage system. 2) The storage

[1] R7 was at the Application category before doing the survey. However, the survey comments on R7 are often related to R1 and R2. Therefore, we move it to the Storage category.

system depends on special storage media, storage drivers and storage software, so that the custodians must have the sufficient components to read this storage medium.

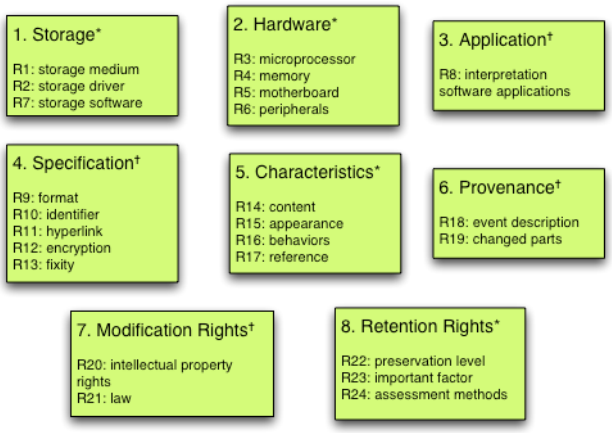

Figure 1 Migration Data Requirements (†. Necessary; *. Conditional)

- **Hardware**: Hardware metadata specify what components are necessary to build a computer system, with which the custodians can read the old storage system and can run the old applications. For example, microprocessor (R3), memory (R4), motherboard (R5), and peripherals (R6) are needed to create a basic computer system. However, the survey respondents commented that these metadata should be preserved when the components of the preservation system are dependent each other. In addition, some of them deemed that just having the name of a given computer generation is enough.
- **Application**: Interpretation software applications (R8) can interpret a technical specification. Most respondents agree that the metadata on R8 must be preserved. It is because any interpretation software application is the key to view and manipulate the preserved digital materials.
- **Specification**: Specification metadata describe techniques used for preserved digital materials. Currently, there are five kinds of techniques that may be used by digital materials, namely format (R9), identifier (R10), hyperlink (R11), encryption (R12), and fixity (R13). Most respondents believe that these requirements are necessary, because the developers of the migration plan must use them to develop a migration solution and compare various migration solutions.
- **Characteristics**: Characteristics metadata define essential facets of a digital material, e.g., content (R14), appearance (R15), behavior (R16) and reference (R17). Using these facets, preservation systems may evaluate the migration results. The respondent argued that whether the preservation system should store characteristics is determined by the existence of two kinds of software

applications: an application that can extract these characteristics and an application that can utilize these characteristics.

- **Provenance**: Provenance metadata describe previous activities on digital materials. It includes the documentation of those activity events (R18) and all changed parts of the preserved digital materials (R19) during the migration. These data are necessary and helpful to improve the trustworthiness of the digital materials.
- **Modification Rights**: Metadata on modification rights specify what kind of migration activity can be carried out. These rights may be intellectual property rights (IPRs, R20) or government law (R21). Hence, in order to keep the migration legal, the custodians must comply with those pre-specified modification rights.
- **Retention Rights**: Retention rights specify a set of preservation rules, which let the custodians to use the same criteria as before. The possible rules could be the preservation level (R22), important factors on characteristics (R23), and assessment methods for migration results (R24). As the theories on R23 and R24 are not mature, the survey respondents deem that these two data requirements are not necessary. However, the respondents believe that R22 is necessary to store.

4 MMET - Migration Metadata Extraction Tool

MMET is implemented by Java and depends on a structural metadata schema called METS [23], which organizes a preservation package including several digital materials. Each METS document contains 7 sub-parts: *metsHrd* describing this METS file, *dmdSec* describing files within this preservation package, *admSec* providing administration information of these files, *fileSec* providing location of these files, *structMap* providing the organizational structure of these files, *structLink* defining hyperlinks between these files, and *behaviorSec* defining software behaviors necessary for viewing or interacting.

Figure 2 illustrates the abstract architecture of MMET. In the architecture, there are an execution part and a specification part. In the execution part, there are 7 tasks that are allocated into four MMET components, namely MMETManager, MMScanner, MMETExtractor and MMETSummary (see Figure 3). In the specification part, there is an external task in which migration specialists should define a set of mapping rules between the preserved metadata and the necessary metadata for migration.

Due to the specification part just has one task. The following description is based on the components of the execution part. The task of the specification part will be mentioned when we introduce MMETExtractor.

MMETManager

MMETManager provides a graphic interface to the custodians. Using this interface, the custodians can select the file folder under which files are going to be

migrated (i.e., Task 1), and view the file situation of this folder and explore the stored metadata for each preserved digital materials in an XML form (i.e., Task 7).

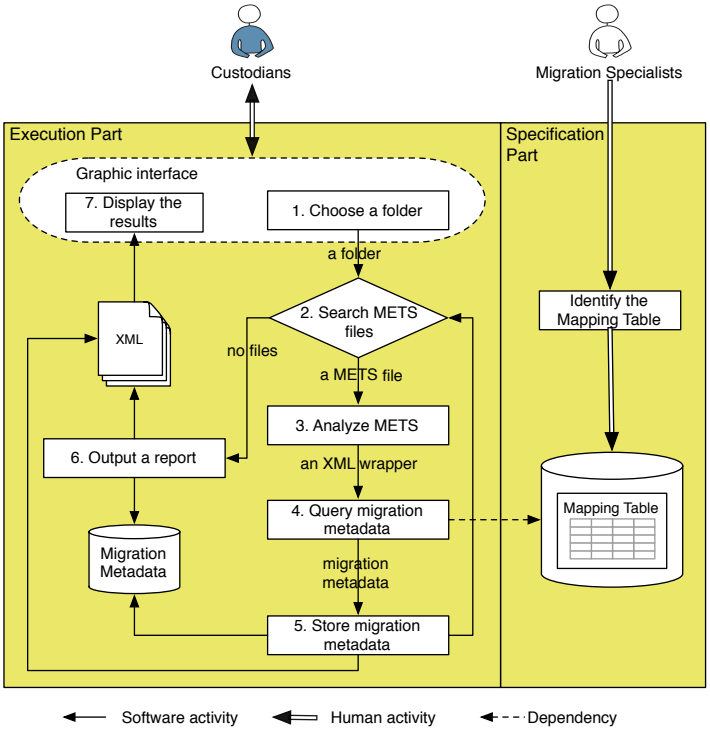

Figure 2 Abstract Architecture of MMET

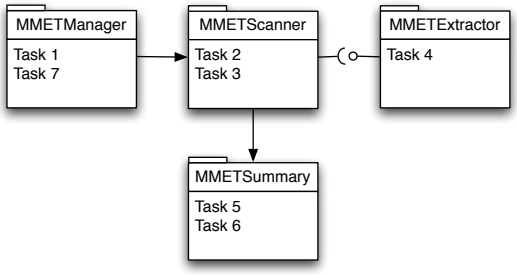

Figure 3 Components of the Execution Part

MMETScanner

MMETScanner carries out Task 2 and Task 3. In Task 2, a set of files including sub-folders is retrieved from a given folder. Then, MMETScanner determines what type each file belongs to. If the type is directory, MMETScanner will go into this sub-folder and do the same task as Task 2 again. If the type is file, MMETScanner will

judge whether it is a METS file or not. Only the METS file is sent to Task 3. Task 2 will be recursively executed until all files have been analyzed.

In Task 3, the METS file would be loaded into memory for analyzing. Firstly, a java library[2] is used to parse METS and extract the METS sub-parts. Secondly, a set of works is deployed to extract the migration metadata. We found that in the METS sub-parts, admSec and fileSec are useful for MMET. In admSec, there are as techMD, rightsMD, digiprovMD, and sourceMD. Each of them contains a wrapper (named mdWrap) or a reference (named mdRef) linking to a XML file. Both the wrapper and the file contain a set of administrative metadata that can provide the migration metadata. In fileSec, a set of files is listed, which are the preserved digital materials. In addition, these files have links that connect to techMD, rightsMD, digiprovMD, and sourceMD. Figure 4 illustrates the relation between fileSec and admSec. Hence, MMET retrieves files from fileSec. Following the links of these files, MMET can go to techMD, rightsMD, digiprovMD, or sourceMD for retrieving the possible wrapper.

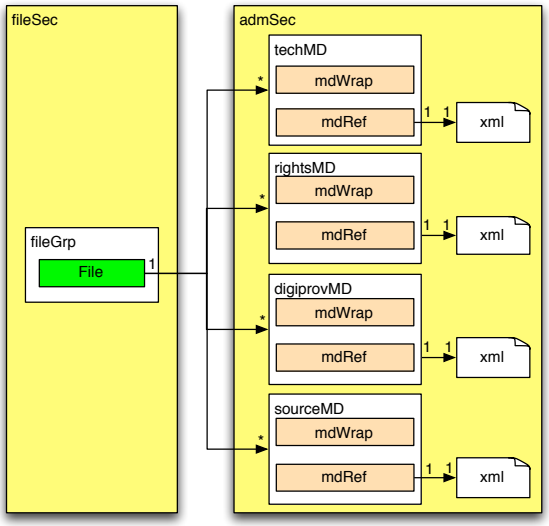

Figure 4 Relationship between FileSec and AdmSec

MMETExtractor

MMETExtractor contains Task 4, which extracts the migration metadata from the XML wrapper generated in Task 3. However, before doing Task 4, MMET requires doing an external task in the specification part, i.e., a migration specialist designs a set of mapping rules between the wrapper schema and the migration data requirements. For instance, in our test dataset, the wrapper uses PREMIS-v1.0 [24] and MIX-v1.0 [25]. PREMIS-v1.0 includes many metadata on the archive package, the single material, rights, events, and agent. Hence, there are many mapping rules for PREMIS-v1.0 (see Table 1). As for MIX-v1.0, it mainly provides the characteristics metadata

[2] From the Australia National University, http://sourceforge.net/projects/mets-api/.

for digital images. This information is just related to R14, so the mapping table has one entry, i.e., MIX-v1.0 -> R14.

Using the mapping rules, Task 4 is able to query the migration metadata. As the test environment has some constraint, we cannot use any database. Hence, the java interface mechanism is used. The interface defines several abstract query operations, whilst the java class for a given wrapper schema provides implementation of those abstract operations. For instance, in our implementation, the PREMIS-v1.0 java class uses the XML path language (XPath) [26] to retrieve the migration metadata. Using this interface mechanism, it is easy for MMET to support any wrapper schema. Finally, Task 4 will transfer the migration metadata to MMETSummary.

Table 1 Mapping Table for PREMIS-v1 and our Migration Data Requirements

Category	**Req.**	**Elements in PREMIS-v1.0**
Storage	R1	• Storage.storageMedium
	R2	• n/a
	R7	• n/a
Hardware	R3	• Environment.hardware.{hwName, hwType, hwOtherInformation}
	R4	• Environment.hardware.{hwName, hwType, hwOtherInformation}
	R5	• Environment.hardware.{hwName, hwType, hwOtherInformation}
	R6	• Environment.hardware.{hwName, hwType, hwOtherInformation}
Application	R8	• Environment. Software.{swName, swVersion, swType, swOtherInformation, swDependency} • CreatingApplication.{creatingApplicationName, creatingApplicationVersion, dateCreatedByApplication, creatingApplicationExtension}
Specification	R9	• objectCharacteristics.format.formatDesignation.{formatName, formatVersion}
	R10	• objectIdentifier.objectIdentifierType
	R11	• relationship.relatedObjectIdentification.relatedObjectIdentifierType • relationship.relatedEventIdentification.relatedEventIdentifierType • linkingEventIdentifier.relatedEventIdentifierType • linkingIntellectualEntityIdentifier.linkingIntellectualEntityIdentifierType • linkingPermisionStatementIdentifier.linkingPermissionStatementIdentifierType
	R12	• objectCharacteristics.inhibitors.{inhibitorType, inhibitorTarget}
	R13	• objectCharacteristics. Fixity.{messageDigestAlgorithm, messageDigestOriginator}
Characteri-stics	R14	• objectCharacteristics.significantProperties
	R15	• objectCharacteristics.significantProperties
	R16	• objectCharacteristics.significantProperties
	R17	• objectCharacteristics.significantProperties
Provenance	R18	• eventType • eventDateTime • linkingAgentIdentifier.{linkingAgentIdentifierType, linkingAgentIdentifierValue, linkingAgentRole}
	R19	• eventOutcomeInformation.{eventOutcome, eventOutcomeDetail}
Modification rights	R20	• permissionStatement.*
	R21	• permissionStatement.*
Retention rights	R22	• preservationLevel
	R23	• n/a
	R24	• n/a

*. All sub-elements of a given element should be provided.

MMETSummary

MMETSummary does Task 5 and Task 6. In Task 5, MMETSummary receives the migration metadata of a given digital material from MMScanner and stores this migration metadata together with other metadata. Like the mapping rules, we have to give up databases and use an in-memory XML data structure, i.e., Document Object Model (DOM), to store the summary information. In addition, MMET will save the migration metadata to an XML file, because the custodians may need to check a single digital material's migration metadata.

When all files in the user-specified folder have been analyzed, Task 2 will invoke Task 6 in MMETSummary to store a report about the overview of this folder to an XML file. In the XML file, the first level contains categories of the migration data requirements. The second level contains instances of the requirements. All the instances are organized in terms of the classification of the requirements. The third and last level contains identifiers of preserved digital materials. Every instance of the requirements would list all identifies of its digital materials. In addition, every identifier has an attribute about the location of an XML file, in which the migration metadata is stored.

5 Experiment Results and Evaluation

In the MMET experiment, we use a number of METS files from the National Library of Norway. The test results show that MMET successfully retrieves much information from the preserved metadata. However, some information still cannot be retrieved as the preservation metadata schema does not have related elements, or the data are not stored into the preservation system at all.

As for the speed aspect, Table 2 summarizes the average times of MMET. The results show that the overall performance increases in a linear growth way. When scanning 1 million METS files, MMET will take nearly 7,8 hours. Hence, we stop the test at the scale of one million, as it will take more than 3 days for MMET to scan 10 million METS files.

Table 2 Performance of MMET (in sec)

Files	Task 3	Task 4	Task 5	Task 6	Other	Overall
$\approx 10^2$	0,72	3,37	0,35	0,31	0,05	4,81
$\approx 10^3$	2,91	22,46	3,81	1,63	0,09	30,91
$\approx 10^4$	23,11	204,78	38,28	14,30	0,52	280,98
$\approx 10^5$	250,65	2044,84	445,36	147,54	5,22	2893,61
$\approx 10^6$	2448,72 (≈40,8 min)	20337,21 (≈5.7 hr)	3898,90 (≈1,1 hr)	1451,18 (≈24,2 min)	66,39 (≈1,1 min)	28202,39 (≈7,8 hr)

We further evaluate MMET by JHOVE. Using JHOVE is because it is often used in preservation systems. The evaluation focuses on the efficiency and the quality and quantity of migration metadata. We use digital books as our testing dataset. Every

page of a digital book is stored in the JPEG-2000 format and the JPEG format, respectively. In addition, the content of the book is extracted by an OCR machine and is stored into an XML file. The associated metadata files are METS files and the output of JHOVE. We find that it is time-consuming for JHOVE to extract characteristics for each digital material. For instance, JHOVE would spend nearly 52 hours for a 303.3 GB dataset, but MMET only needs 78 minutes for the same dataset. As the characteristics extraction function of JHOVE spends too much time, we have to use the audit function of JHOVE (written JHOVE Audit), which validates file formats and creates an inventory about the file system. We tested JHOVE Audit and MMET. Table 3 illustrates the evaluation results. JHOVE Audit and MMET have similar speeds, but JHOVE is very slow.

Table 3 Performances of MMET, JHOVE and JHOVE Audit* (in hr)

Dataset	MMET	JHOVE Audit	JHOVE
303,3 GB	≈ 1,3	≈ 1,1	≈ 52,0
606,6 GB	≈ 2,6	≈ 2,3	n/a
909,9 GB	≈ 3,9	≈ 3,3	n/a
1213,2 GB	≈ 5,4	≈ 4,5	n/a

*. JHOVE means JHOVE does the characteristics extraction function, whilst JHOVE Audit means JHOVE just does the audit function.

As for the quality and the quantity of retrieved metadata, JHOVE Audit creates few metadata. It just reports the validity statue, format types in the MIME classification, and the number of files for a given format and folder. For instance, for the 303.3 GB dataset, JHOVE Audit reports that all files are valid and there are 4 kinds of formats, i.e., image/jp2, image/jpg, text/plain with the US-ASCII charset, and text/plan with the UTF-8 charset[3]. Compared again the real situation, we found this information is not accurate. JHOVE Audit recognizes most of XML files using UTF-8 as US-ASCII.

JHOVE creates more metadata than JHOVE Audit. For each file, JHOVE shows not only the validity and the MIME format type, but also it retrieves metadata embedded in the file and generates characteristics metadata based on the content. For instance, JHOVE use MIX-v1.0 to store characteristics of images.

MMET provides more metadata than JHOVE Audit and JHOVE. In the MMET report, there are many metadata about storage, software, format, identifier, reference, fixity, preservation level, the schema for wrapping provenance, and the schema for wrapping characteristics. As for the format metadata, MMET reports JPEG2000, JPEG-1.01, and XML-1.0. This is the same as the real situation. Therefore, for the quality and the quantity of the outputted metadata, MMET is the best in our evaluation.

[3] text/plain with the US-ASCII or UTF-8 charset refers to a XML format. Since our test environment has no Internet, the XML module of JHOVE cannot be used.

6 Further Discussion

There are two methods to obtain information for a migration plan design. The first method is called file-based solution, which directly analyzes digital materials, like JHOVE. The second method is named metadata-based solution, which retrieves information from the preserved metadata, like MMET. At different time points, these two methods can play different roles. For instance, when a digital material is inserted into the preservation system, there are few metadata. Hence, the metadata-based solution will not work at all. The file-based solution should be used.

However, in the preservation period, the metadata-based solution works better than the file-based solution. The file-based solution can only be used to do some simple functions, such as identifying formats. This is because 1) the file-based solution is slow when it realizes a complex function, e.g., characteristics extraction; 2) the extracted metadata may not be the same as the real situation; and 3) many redundancy files, which were ever used but are not important now, may be involved in the calculation of the file-based solution.

The metadata-based solution plays well in the preservation period. It can retrieve many metadata, and the retrieved metadata are more accurate than the file-based solution. Moreover, the metadata-based solution does not need to access the preserved digital materials when the custodians design a migration plan. This advantage is helpful to increase security of the preservation system, and makes it possible for the preservation system to outsource the migration plan design job. For instance, a third-party institution can assess risks in the preservation system and design corresponding solutions. However, the metadata-based solution has some limitations: 1) the quality and the quantity of preserved metadata will affect the migration metadata; and 2) a manual intervention is involved, e.g., defining mapping rules.

However, the speed is a big challenge for both the metadata-based solution and the file-based solution. In our test, the 1213,2 GB dataset contains 1280 digitized books with 57380 pages in total. MMET needs around 5,4 hours to retrieve the metadata, and JHOVE Audit needs 4,5 hours. However, large preservation systems, such as national libraries or national archives, have hundreds and thousands books. When all these books are digitized, it may take many days or months to retrieve metadata. In this situation, both the metadata-based solution and the file-based solution are bad. The possible solutions are 1) parallel computing technique should be used in the metadata-based solution and the file-based solution, and 2) the management task for the metadata should be transferred from the application level to the system level. For example, there exist a preservation-aware storage in [27], in which some metadata can be added.

7 Conclusion

Migration is a time-consuming and expensive task for preservation systems. When the custodians design a migration plan, they ask to obtain necessary and sufficient metadata. In terms of our previous study on the migration data requirements, we find

that the current solutions just provide a part of necessary metadata. Hence, MMET is designed to analyze the preserved metadata and retrieve related metadata from them. In the experiment, MMET outputs many metadata for migration. However, in terms of the migration data requirements, some of metadata still cannot be retrieved, as the preservation system does not store them at all. For the performance aspect, under the almost same time, MMET can obtain the overview of the file system and metadata for every digital material, whereas JHOVE just generates the overview.

Acknowledgements. Research in this paper is funded by the Norwegian Research Council and our industry partners under the LongRec project. We would also thank our partners of LongRec, especially the National Library of Norway for providing experiment environment and technique supports.

References

1. The Consultative committee for Space Data Systems: The Reference Model for an Open Archival Information System (OAIS) (2002).
2. Lee, K.H., Slattery, O., Lu, R., Tang, X., McCrary, V.: The State of the Art and Practice in Digital Preservation. Journal of Research of the National Institute of Standards and Technology 107(1), 93-106 (2002).
3. Thibodeau, K.: Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years. CLIR Reports, Conference Proceedings of The State of Digital Preservation: An International Perspective (2002).
4. Wheatley, P.: Migration–a CAMiLEON discussion paper. Ariadne 29(2) (2001).
5. Granger, S.: Emulation as a Digital Preservation Strategy. D-Lib Magazine 6(10) (2000).
6. Lorie, R.A.: A methodology and system for preserving digital data. In: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, Portland, Oregon, USA 2002, pp. 312-319. ACM (2002).
7. Borghoff, U., Rödig, P., Schmitz, L., Scheffczyk, J.: Migration: Current Research and Development. In: Long-Term Preservation of Digital Documents. pp. 171-206. Springer Berlin Heidelberg (2006).
8. Stanescu, A.: Assessing the durability of formats in a digital preservation environment - The INFORM methodology. OCLC Systems & Services: International Digital Library Perspectives 21(1), 61-81 (2005).
9. Li, C., Zheng, X.H., Meng, X., Wang, L., Xing, C.X.: A methodology for measuring the preservation durability of digital formats. Journal of Zhejiang University - Science C 11(11), 872-881 (2010).
10. Strodl, S., Becker, C., Neumayer, R., Rauber, A.: How to choose a digital preservation strategy: evaluating a preservation planning procedure. In: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, Vancouver, BC, Canada 2007, pp. 29-38. ACM (2007).
11. Becker, C., Kulovits, H., Guttenbrunner, M., Strodl, S., Rauber, A., Hofman, H.: Systematic planning for digital preservation: evaluating potential strategies and building preservation plans. International Journal on Digital Libraries 10(4), 133-157 (2009).
12. Thaller, M., Heydegger, V., Schnasse, J., Beyl, S., Chudobkaite, E.: Significant Characteristics to Abstract Content: Long Term Preservation of Information. In:

Christensen-Dalsgaard, B., Castelli, D., Ammitzbøll Jurik, B., Lippincott, J. (eds.) Research and Advanced Technology for Digital Libraries, vol. 5173. Lecture Notes in Computer Science, pp. 41-49. Springer Berlin / Heidelberg (2008).
13. ExifTool. http://www.sno.phy.queensu.ca/~phil/exiftool/.
14. Tika. http://tika.apache.org/.
15. DROID. http://sourceforge.net/apps/mediawiki/droid/index.php?title=Main_Page.
16. PRONOM. http://www.nationalarchives.gov.uk/pronom/.
17. Fido. https://github.com/openplanets/fido.
18. Abrams, S.L.: The role of format in digital preservation. Vine 34, 49-55 (2004).
19. Anderson, R., Frost, H., Hoebelheinrich, N., Johnson, K.: The AIHT at Stanford University: Automated preservation assessment of heterogeneous digital collections. D-Lib magazine 11, 12 (2005).
20. Marketakis, Y., Tzanakis, M., Tzitzikas, Y.: PreScan: towards automating the preservation of digital objects. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems 2009, pp. ACM--60:411 (2009).
21. File Information Tool Set (FITS). http://code.google.com/p/fits/.
22. Luan, F., Mestl, T., Nygård, M.: Quality Requirements of Migration Metadata in Long-Term Digital Preservation Systems. In: Sánchez-Alonso, S., Athanasiadis, I.N. (eds.) Metadata and Semantic Research, vol. 108. Communications in Computer and Information Science, pp. 172-182. Springer Berlin Heidelberg, (2010).
23. McDonough, J.: METS: standardized encoding for digital library objects. International journal on digital libraries 6(2), 148-158 (2006).
24. PREMIS Data Dictionary for Preservation Metadata 1.0. In. The PREMIS Editorial Committee, (2005).
25. ANSI/NISO Z39.87 - Data Dictionary - Technical Metadata for Digital Still Images. In. ANSI/NISO, (2006).
26. Clark, J., DeRose, S.: XML Path Language (XPath) version 1.0 w3c recommendation. In: Technical Report REC-xpath-19991116. World Wide Web Consortium, (1999).
27. Factor, M., Naor, D., Rabinovici-Cohen, S., Ramati, L., Reshef, P., Satran, J., Giaretta, D.L.: Preservation DataStores: Architecture for Preservation Aware Storage. In: Mass Storage Systems and Technologies, 2007. MSST 2007. 24th IEEE Conference on 2007, pp. 3-15 (2007).