SEMEXSS — A Rule-Based Semantic Metadata Extraction System for Spreadsheets

Somchai Chatvichienchai

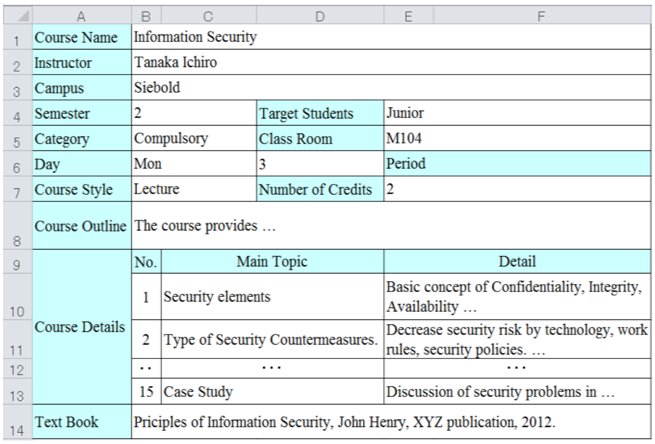
***Abstract*—Spreadsheets store not only routine data but also valuable information for organization administration and planning. Finding the spreadsheets that fit users' needs from disparate repositories is becoming increasingly important. Semantic metadata is known as metadata that describes contextually relevant about content which is based on an industry-specific or enterprise-specific custom metadata model. Therefore, semantic metadata is used by many document management systems and search systems to search documents of organizations. However, due to limitation of current metadata extraction methods, semantic metadata extraction cannot be done automatically in many cases. The objective of this paper is to propose a novel system called *SEMEXSS* that can extract semantic metadata automatically from spreadsheets by metadata extraction rules. The extraction rules are automatically generated by the program that reads a sample spreadsheet whose semantic metadata is defined by users via a user interface of spreadsheet software. Experiment is done to investigate time complexity of metadata extraction of the system.**

***Index Terms*—Metadata, generating, schema, semantic, XML.**

I. Introduction

Spreadsheet programs, such as Microsoft Excel [1], Lotus 1-2-3 [2] and Calc [3], are used by millions of users as a routine all-purpose data management tool. Spreadsheets store not only routine data but also valuable information for organization administration and planning such as financial statements, marketing analysis reports, etc. As more and more spreadsheets become electronically available, finding the spreadsheets that fit users' needs from disparate repositories is becoming increasingly important. Current software tools, such as Copernic Desktop Search [4], X1 Professional Client [5], which perform a search using file names, file content and syntactic metadata (such as file creation date, etc.), are not sufficient to handle the above problems. A new concept known as semantic metadata is paving the way to finally realize the full value of information. Semantic metadata is known as metadata that describes contextually relevant or domain-specific information about content which is based on an industry-specific or enterprise-specific custom metadata model or ontology.

Several tools, such as Metadata Miner Pro [6], SemreX [7], have been proposed to automatically extract metadata from documents. However, spreadsheets hold characteristics that are different from documents of other types. A spreadsheet holds a set of sheets which are viewed as grids of cells. A cell or a range of cells may contain a value or a formula. The value of a cell or a range may have semantic relationship with the values of the adjacent cells or ranges. Furthermore, each sheet may contain tables which are viewed as two-dimensional arrays. This semantic relationship can be used to define semantic metadata inside a spreadsheet. Unfortunately, current metadata extraction tools did not pay attention to these characteristics. To handle this important issue, my previous work [8] proposed a novel method that extracts semantic metadata from a spreadsheet whose layout is similar to that of a registered template whose semantic metadata is previously defined. However, this method has two drawbacks. The first is that the metadata definition of the template is difficult for end users to understand since it is based on mapping from spreadsheet's cells to metadata schema elements. The second is that high overhead in comparing layout between registered templates and a spreadsheet whose metadata will be extracted.



	A	B	C	D	E	F
1	Course Name	Information Security				
2	Instructor	Tanaka Ichiro				
3	Campus	Siebold				
4	Semester	2		Target Students	Junior	
5	Category	Compulsory		Class Room	M104	
6	Day	Mon		3	Period	
7	Course Style	Lecture		Number of Credits	2	
8	Course Outline	The course provides …				
9	Course Details	No.	Main Topic		Detail	
10		1	Security elements		Basic concept of Confidentiality, Integrity, Availability …	
11		2	Type of Security Countermeasures.		Decrease security risk by technology, work rules, security policies. …	
12		‥	…		…	
13		15	Case Study		Discussion of security problems in …	
14	Text Book	Priciples of Information Security, John Henry, XYZ publication, 2012.				

Fig. 1. An example of a spreadsheet presenting course information.

Fig. 1 shows a spreadsheet presenting an example of course information. In order to enable users to search this spreadsheet by semantic metadata-based query, a metadata crawler has to generate proper metadata and its semantic definition from this spreadsheet. Fig. 2 illustrates an example of the semantic metadata of this spreadsheet which is outputted in XML format. By storing the semantic metadata set as XML data, users can define XML tag to denote the meaning of the data enclosed by the tag. For example, *table1* element is defined to contain *mainTopic* and *detail* elements in order to enable users to define search queries based on course details of the same day. The semantic metadata will be sent to indexer process to product search index for the spreadsheet of Fig. 1.

The objective of this research is to propose *a Rule-based Semantic Metadata Extraction System for Spreadsheets*

Manuscript received September 12, 2014; revised February 2, 2015.

Somchai Chatvichienchai is with the Department of Information and Media Studies, University of Nagasaki, 851-2195 Japan (e-mail: somchaic@sun.ac.jp).

DOI: 10.7763/IJCTE.2016.V8.1027

(*SEMEXSS*, for short) that can solve the above two drawbacks. In order to simplify semantic metadata extraction problem, this work focuses on spreadsheet collections which are categorized by layout similarity. The main idea of semantic metadata extraction is that system manager, who is in charge of managing metadata-based search system, selects a sample spreadsheet from a spreadsheet collection of the same category. She defines a metadata schema presenting the metadata set for the spreadsheet collection. In order to locate metadata, she maps from metadata schema elements to cells or cell ranges containing semantic metadata of the sample spreadsheet.

```
<?xml version="1.0" encoding="UTF-8"?>
<courseInfo>
    <single>
      <courseName> information security</courseName>
      <instructor> tanaka ichiro </instructor >
      <campus> siebold </campus>
      <semester> second </semester >
      <targetStudents > junior </targetStudents>
      <category> compulsory </category >
      <classRoom> m104 </classRoom >
      <day> mon </day >
      <timePeriod> 3 </timePeriod >
      <style> lecture </style >
      <credit> 2 units </credit>
      <outline> this course provides …</outline>
      <textBook> priciples of information security,
            john henry, xyz publication, 2012. </textBook>
    </single>
    <table1>
      <mainTopic> security elements </mainTopic>
      <detail> basic concept of …</detail>
      <unitPrice> 1000 </unitPrice>
    </table1>
    <table1>
      <mainTopic> type of security countermeasures. </mainTopic>
      <detail> decrease security risk…</detail>
    </table1>
      …
    <table1>
      <mainTopic> case study </mainTopic>
      <detail> discussion of security …</detail>
    </table1>
</courseInfo >
```

Fig. 2. An example of semantic metadata generated from the spreadsheet of Fig. 1.

The first drawback of my previous work is solved by developing a program that reads the sample spreadsheet bound with metadata schema and outputs category justification rules and metadata extraction rules for the spreadsheet collection whose category is the same that of the sample spreadsheet. The second drawback of my previous work is solved by developing a program that justifies the category of given spreadsheet by looking up category justification rules of each category and generates XML-based metadata for a given spreadsheet by metadata extraction rules of the relevant category.

The rest of the paper is organized as follows. The second section introduces basic concepts of metadata, XML and XML schema. Issues in generating semantic metadata of a spreadsheet are discussed in the third section. The architecture of the proposed system is presented in the fourth section. The fifth section presents an algorithm extracting semantic metadata and experiment results. The related work is discussed in the sixth section. Finally, the last section concludes this paper and future work.

II. Basic Concept

A. Metadata

Metadata is data about data, more specifically a collection of key information about a particular content, which can be used to facilitate the understanding, use and management of data. Metadata is classified into two following categories.

1) Syntactic metadata

This metadata describes file attributes (such as file size, file path or file creation date, etc.). It does not provide a level of understanding about what the spreadsheet says or implies.

2) Semantic metadata

This metadata describes contextually relevant or domain-specific information about content which is based on an enterprise-specific custom metadata model or ontology. For example, if the content is from the business domain, the relevant semantic metadata might be company name, industry, sector, location, etc.

Since syntactic metadata can be easily obtained by using application program interface of file system of the operating system, syntactic metadata extraction is not in scope of this paper. This paper focuses on semantic metadata which is used to define search condition. For readability, in this paper the term “metadata” denotes “semantic metadata”.

B. XML and XML Schema

XML [9] provides a way to describe structured data. Unlike HTML tags, which are originally used to define appearance of data, XML tags are used to define the data types and structure of the data itself. XML uses a set of tags to delineate elements of data. Each element encapsulates a piece of data that may be very simple or very complex. In this paper, XML is used to describe the values and semantic of metadata.

XML schema [10] is a document used to define and validate the content and structure of XML data. It is similar to a relational database schema defines and validates the tables, columns, and data types that make up a relational database. XML Schema defines and describes certain types of XML data by using the XML Schema definition language (XSD). In this paper, an XML schema is used to describe a metadata schema which defines the relationships between metadata elements, and the syntax and the optionality (obligation level) of values. For readability, a metadata schema is presented as a hierarchical tree.

As shown in the left hand side of Fig. 3, metadata schema of course information consists of *category_identifier* and metadata elements. *Category_identifier* element consists of keyword elements each of which defines the text string that is in a spreadsheet’s cell and is used to identify the category of the spreadsheet. Metadata element consists of single and table elements. Single element defines the metadata whose cardinality is one. A table element defines the metadata stored in a table of the spreadsheet.

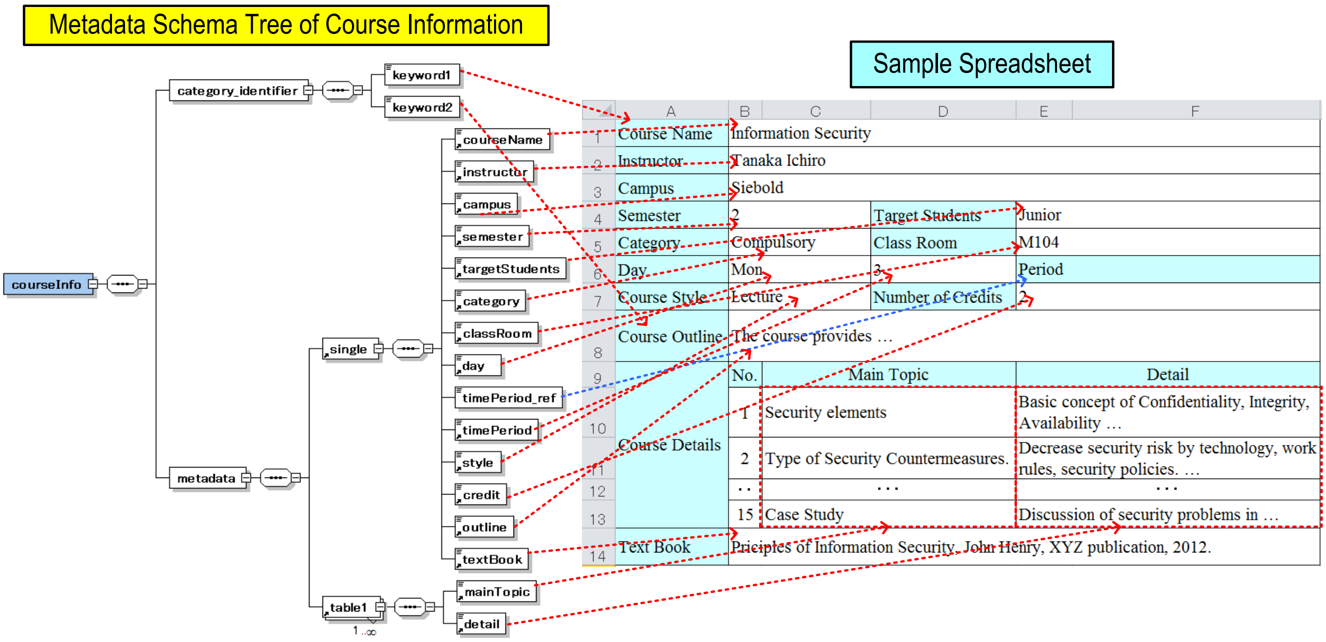


Fig. 3. A sample spreadsheet bound with metadata schema of course information.

III. Issues in Extracting Semantic Metadata from a Spreadsheet

A. Justification of Category of a Spreadsheet

Since users generally create a spreadsheet of the same category from the same template, spreadsheets of the same category tend to have the same layout. Based on this assumption, defining metadata extraction rules for spreadsheets of the same category is an effective approach. Before metadata extraction, it is necessary to justify category of a spreadsheet to select the corresponding metadata extraction rules. This paper introduces a spreadsheet category justification method which is based on layout properties of the spreadsheet. This method is similar to that is done by human visual inspection. Suppose that spreadsheet of Fig. 3 is the sample spreadsheet selected from spreadsheet collection of course information category. The category of the sample spreadsheet is defined by the existence of string "Course Name" of cell A1 and string "Course Outline" of cell A8. This definition is done by mapping keyword1 element to cell A1 and keyword2 element to cell A8 of the spreadsheet. In case of Microsoft Excel, the mapping from metadata schema elements to spreadsheet's cells is done by XML source task pane [11] of Excel.

B. Definition of Semantic and Location of Metadata of a Spreadsheet

The semantic of the metadata of the spreadsheet of Fig. 3 is defined by mapping from schema elements of single element and *table1* element to corresponding cells or ranges of the spreadsheet. For example, targetStudents element is mapped to cell E4. This mapping states that cell E4 stores metadata of *targetStudent* class. However, this definition is not flexible since it is not correct if users insert a row before E column. In order to solve this problem, this paper defines a text string that is adjacent to metadata to locate the metadata. The text strings used to locate metadata are called metadata-identifiers (m-identifiers, for short). The location relationship between metadata and its m-identifier varies on metadata type. In this paper, the location relationship between metadata and its m-identifier can be classified into the following three types.

1) **After m-identifier:** In this type, metadata is after its m-identifier. For example, string "Instructor" of cell A2 of Fig. 3 is m-identifier for string "Tanaka Ichiro" of cell range(B2:F2).
2) **Before m-identifier:** In this type, metadata is before its m-identifier. For example, string "Period" of cell range(E6:F6) of Fig. 3 is m-identifier for string "3" of cell D6.
3) **Under m-identifier:** In case metadata is table data, m-identifier of metadata is usually the column header. Therefore, the metadata of this type is under its m-identifier. For example, text string "Security elements" of cell range(C10:D10) of Fig. 3 is metadata of *mainTopic* class. It is under string "Main Topic" of cell range(C9:D9) which is its m-identifier.

Since "After m-identifier" is mostly found in many spreadsheet categories, system manager defines it as the default location relationship between metadata and its m-identifier. In case location relationship between some metadata and its m-identifier is different from the default local relationship, system manager has to define metadata schema elements presenting those m-identifiers and maps the schema elements with those m-identifiers. For example, *timePeriod_ref* denotes the schema element mapped to m-identifier of metadata of *timePeriod* class. Based on the sample spreadsheet of Fig. 3, the location relationship between metadata of *timePeriod* class and its m-identifier is "Before m-identifier".

IV. System Architecture

As shown in Fig. 4, the system consists of the following four processes: Metadata Schema Definition & Binding, Metadata Extraction Rule Definition, Rule Verification & Modification, and Metadata Extraction.

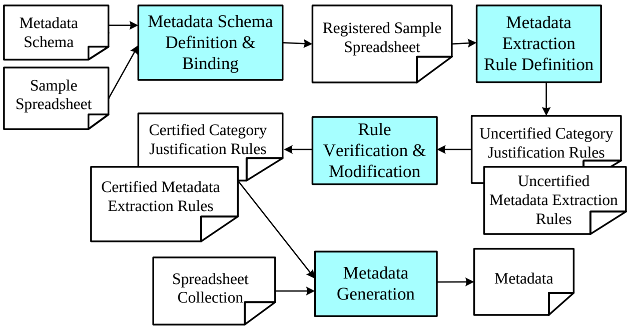

Metadata Schema
Sample Spreadsheet
Metadata Schema Definition & Binding
Registered Sample Spreadsheet
Metadata Extraction Rule Definition
Certified Category Justification Rules
Rule Verification & Modification
Uncertified Category Justification Rules
Certified Metadata Extraction Rules
Uncertified Metadata Extraction Rules
Spreadsheet Collection
Metadata Generation
Metadata

Fig. 4. System architecture.

A. *Metadata Schema Definition and Binding Process*

In this process, system manager selects a sample spreadsheet from a spreadsheet collection of a category. She defines a metadata schema for the spreadsheets of selected category. The class and location of metadata are defined by mapping the corresponding XML element of metadata schema to a cell or a range of the sample spreadsheet that stores the metadata. In this paper, a sample spreadsheet whose cells or ranges are bound with schema elements of metadata schema is called registered sample spreadsheet. In case of Excel, Excel includes metadata schema definition into the registered sample spreadsheet.

B. *Metadata Extraction Rule Definition Process*

In order to enable metadata crawler to justify spreadsheet category and cells/ranges storing metadata, system manager has to define *category justification rules* and *metadata extraction rules* of spreadsheets of each category.

A category justification rule is a 5-tuple of the form:

(*rule-id*, *category*, *m-class*, *attribute-set*, *identifier*),

where *rule-id* is a rule identifier, *category* is a spreadsheet category, *m-class* is an element name of metadata schema of category, *identifier* denotes a text string used to identify the category of the spreadsheet, and *attribute-set* is an attribute set (such as location, font style and size, etc.) of *identifier*. Consider an example of category justification rules shown in Table I. Rule p_1 states that a spreadsheet of course information category has a string "Course Name" which is located at cell A1. Note that p_1 is derived by analyzing the mapping between cell A1 and schema element keyword1 of course information's metadata schema. However, *attribute-set* argument of the category justification rule will be defined by system manage at the next process.

A metadata extraction rule is a 7-tuple of the form:

(*rule-id*, *category*, *m-group*, *m-class*, *loc*, *m-identifier*, *trans*),

where *rule-id* is a rule identifier, *category* is a spreadsheet category, *m-group* is the name of an element of metadata schema of *category*, *m-class* is the name of a child element of *m-group*, *m-identifier* denotes a text string used as an metadata-identifier, *loc* is location of the cell or range storing metadata with respect to the location of *m-identifier*, and *trans* denotes the method that converts metadata into desired format.

These rules define locations and classes of metadata of a spreadsheet. Location of metadata is defined by referring the location of an m-identifier. Table II depicts an example of metadata extraction rules of course information category. For example, r_9 states that metadata of *timePeriod* class is located before the string "Period". This rule is derived by analyzing the locations of cells of registered sample spreadsheet mapped with *timePeriod_ref* and *timePeriod* schema elements. The format transformation "number → string" of *timePeriod* instructs metadata crawler to eliminate irrelevant symbols (such as comma, etc.) from the metadata. By this way, some parts of category justification rules and metadata extraction rules are generated from the given a registered sample spreadsheet bound with a metadata schema. However, *trans* argument of the metadata extraction rule will be defined by system manager at the next process.

C. *Rule Verification and Modification Process*

In this process, system manager verifies category justification rules and metadata extraction rules outputted by the previous process. She may insert or modify category justification rules and metadata extraction rules to achieve the objective of metadata extraction. Furthermore, system manager should define format transformation of each metadata in order to enable search engine compute search result correctly.

TABLE I: AN EXAMPLE OF CATEGORY JUSTIFICATION RULES

Category	Rule-id	M-Class	Attribute Set	Identifier
Course Information	p_1	keyword1	Loc: A1	"Course Name"
Course Information	p_2	keyword2	Loc: A8	"Course Outline"
Invoice	p_3	keyword1	Loc: Range(A1:B1)	"University of Nagasaki"
Invoice	p_4	keyword2	Loc: D1	"Invoice"

TABLE II: AN EXAMPLE OF SEMANTIC METADATA DEFINITION OF COURSE INFORMATION CATEGORY

Category	Rule-id	M-Group	M-Class	Location	m-identifier	Format Transformation
Course Information	r_1	single	courseName	after	"Course Name"	string → lower case string
	r_2	single	instructor	after	" Instructor"	string → lower case string
	...	...	...	...	...	...
	r_9	single	timePeriod	before	" Period"	number → string
	...	...	...	...	...	...
	r_{14}	table1	mainTopic	under	"Main Topic"	string → lower case string
	r_{15}	table1	detail	under	"Detail"	string → lower case string

D. Metadata Generation Process

The system checks properties of a given spreadsheet from the given spreadsheet, the certified category justification rule set and the metadata extraction rule set. If properties of the given spreadsheet match with all category justification rules of a registered category, the system will employs metadata extraction rules of that category to generate metadata from the given spreadsheet.

V. Metadata Extraction Algorithm

This section introduces the *MetadataExt* algorithm which extracts semantic metadata from a given spreadsheet. The algorithm is described as follows.

MetadataExt (S, T, P, R, Flag, Result).

Input:

- S be a spreadsheet from which metadata will be generated,
- $T = \{t_1, t_2, .. , t_q\}$be the metadata schema set where t_i is a metadata schema of category i and $1\leq i \leq q$,
- $P = \{p_1, p_2, .. , p_m\}$ is the category justification rule set where $p_j = (rule\text{-}id_j, category_j, m\text{-}class_j, attribute\text{-}set_j, identifier_j)$ and $1\leq j \leq m$ (Note that category justification rules are sorted by the category), and
- $R = \{r_1, r_2, .. , r_w\}$ is the metadata extraction rule set where $r_k = (rule\text{-}id_k, category_k, m\text{-}group_k, m\text{-}class_k, location_k, m\text{-}identifier_k, trans_k)$ and $1\leq k \leq w$. (Note that metadata extraction rules are sorted by the value of *category* argument).

Output:

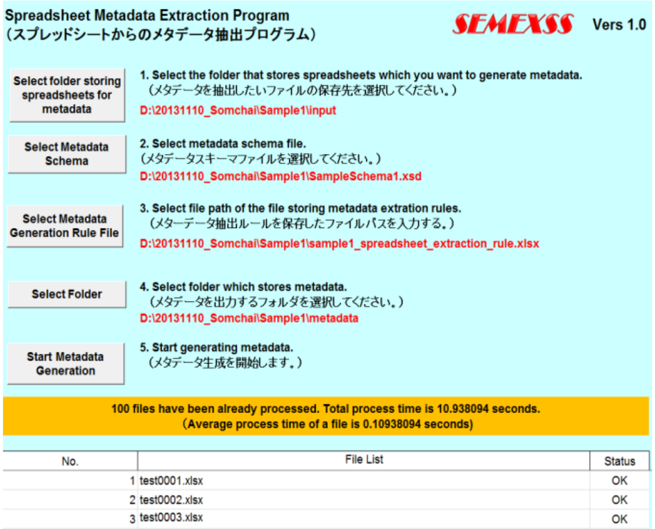
- $flag$ = 'yes' if metadata of S is correctly generated. Otherwise, $flag$ = 'no'.
- *Result* stores metadata of S.

Process:

```
$Flag$ = 'no'; $i$ = 1
Do the following until $i = m$ and $flag$ = 'yes'
If $identifier_i$ of $p_i \in P$ is found in $S$'s range whose attributes are satisfied by $attribute\text{-}set_i$ then /* check other justification rules of the same category */
If each $p_k \in P$ and $category_k = category_i$ and $i \neq k$ and $identifier_k$ of $p_k$ is found in $S$'s range whose attributes are satisfied by $attribute\text{-}set_k$ then
$Flag$ = 'yes'
Else /* check other categories */
If there exists $p_j \in P$ and $j > i$ and $category_i \neq category_j$ Then
$i = j$
Else
return
Else
$i = i + 1$
End Do
Let $R' = \{r'_1, r'_2, .. , r'_v\}$ be a subset of $R$ where $r'_j = (rule\text{-}id_j, category_j, m\text{-}group_j, m\text{-}class_j, location_j, m\text{-}identifier_j, trans_j)$ and $category_j = category_i$, and $1\leq j \leq v$.
$j = 1$
Do the following until $j = v$ and $Flag$ = 'no'
If $m\text{-}identifier_j$ of $r'_j \in R'$ is found in $S$'s range then
If $m\text{-}group$ is 'single' then /* Process non-table data */
Bind $t_i$'s schema element whose name is $m\text{-}class_j$ to the closest range storing a string whose location w.r.t. that of $m\text{-}identifier_j$ is $loc_j$.
Else
Bind $t_i$'s schema element whose name is $m\text{-}class_j$ to the closest ranges storing strings that are under that of $m\text{-}identifier_j$.
Else
$Flag$ = 'no'
$j = j + 1$
End Do
If $Flag$ = 'yes' then
Output the data of $S$ which is bound to $t_i$ as XML data into $Result$.
Return
```

Time Complexity of MetadataExt

Consider the first Do statement of the algorithm. This Do statement will be executed at most m times (where m is the number of category justification rules) to justify the category of the inputted spreadsheet. Time complexity of justify the category of inputted spreadsheet is $O(m)$. Consider the second Do statement of the algorithm. This statement will be execute at most w times (where w is the number of metadata extraction rules) to generate metadata from the inputted spreadsheet. Let c is number of spreadsheet cells containing text string. Based on instructions between the second Do and End Do statements, program compares metadata identifiers of the metadata extraction rules with spreadsheet cells containing text strings. Time complexity of this process is $O(cw)$. Therefore time complexity of *MetadataExt* is $O(m+cw)$. However, time complexity of justifying the category of the inputted spreadsheet is very trivial *w.r.t.* complexity of generating metadata by metadata extraction rules. Therefore, the approximate time complexity of *MetadataExt* is $O(cw)$.


Select folder storing spreadsheets for metadata
Select Metadata Schema
Select Metadata Generation Rule File
Select Folder
Start Metadata Generation

Spreadsheet Metadata Extraction Program
（スプレッドシートからのメタデータ抽出プログラム）

SEMEXSS Vers 1.0

1. Select the folder that stores spreadsheets which you want to generate metadata.
（メタデータを抽出したいファイルの保存先を選択してください。）
D:\20131110_Somchai\Sample1\input

2. Select metadata schema file.
（メタデータスキーマファイルを選択してください。）
D:\20131110_Somchai\Sample1\SampleSchema1.xsd

3. Select file path of the file storing metadata extration rules.
（メタ―データ抽出ルールを保存したファイルパスを入力する。）
D:\20131110_Somchai\Sample1\sample1_spreadsheet_extraction_rule.xlsx

4. Select folder which stores metadata.
（メタデータを出力するフォルダを選択してください。）
D:\20131110_Somchai\Sample1\metadata

5. Start generating metadata.
（メタデータ生成を開始します。）

100 files have been already processed. Total process time is 10.938094 seconds.
(Average process time of a file is 0.10938094 seconds)

No.	File List	Status
1	test0001.xlsx	OK
2	test0002.xlsx	OK
3	test0003.xlsx	OK

Fig. 5. Screenshot of the metadata generation program.

Fig. 5 shows the screenshot of a program that is a prototype of *MetadataExt*. The objective of development of this program is to confirm the correctness and time complexity of *MetadataExt*. In order to save development time, this program is developed by using Visual Basic Application (VBA) [12] of Excel. As shown in Fig. 5, users define the following information.

- The folder storing spreadsheets whose metadata will be generated,
- The file storing spreadsheet category justification rules and metadata extraction rules,
- The metadata schema file, and
- The folder storing the outputted metadata.

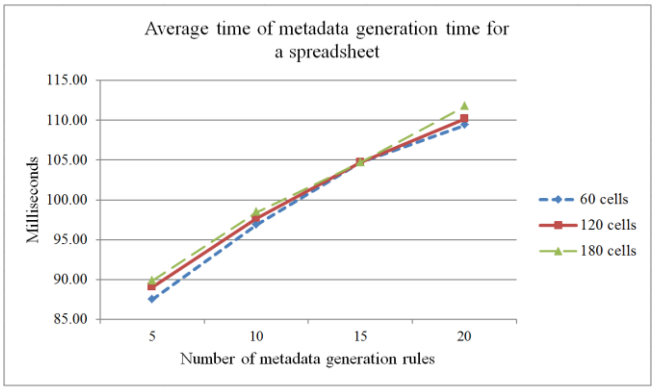
The file list of the inputted spreadsheets and their process status are outputted in the table area of the program. Number of processed spreadsheets and total process time are shown after the program finished. Fig. 6(a) shows average time of metadata generation time for an inputted spreadsheet where

- Number of spreadsheet category justification rules is fixed to 2,
- Numbers of metadata extraction rules are 5, 10 ,15, 20,
- Numbers of spreadsheet cells storing text strings are 60, 120, 180, and
- The average time of metadata generation an inputted spreadsheet is calculated from the result of testing 100 inputted spreadsheets.

	Number of metadata generation rules			
Number of cells storing text strings of an inputted spreadsheet	5	10	15	20
60 cells	87.50	96.88	104.69	109.38
120 cells	89.06	97.66	104.69	110.16
180 cells	89.84	98.44	104.69	111.72

Average time (ms) of metadata generation for a spreadsheet.

(a)



(b)

Fig. 6. Experiment result of the metadata generation.

In order to eliminate the impact of performance of disk units on the experiment result, file I/O time is not included in the average times. Fig. 6(b) depicts a graph comparing average times of metadata generation shown in the table of Fig. 6(a). The graph shows that variation of number of spreadsheet cells containing text strings has a little impact on the metadata generation time w.r.t. variation of number of the metadata extraction rules. Since a spreadsheet category justification rule pinpoints location of cell to be check, the time of justifying the category of an inputted spreadsheet is very trivial w.r.t. the time of generating a metadata. Therefore, it can be confirmed that the approximate time complexity of *MetadataExt* is $O(cw)$ where c is number of spreadsheet cells containing text string, and w is the number of metadata extraction rules.

VI. Related Work

Existing automated metadata extraction approaches can be divided into two main approaches: machine learning approach and rule-based approach. The basic concept of the machine learning approach is to learn the relationship between the input and output of samples and then predict new data. Although this approach has good adaptability, it must be trained from samples. Learning techniques including Support Vector Machine (SVM) [13], [14] and Hidden Markov Model (HMM) [15] have been employed with promising results but to relatively homogeneous document sets. The experiments [16], which are conducted by using SVM and HMM techniques, suggest a significant decline in effectiveness as the heterogeneity of the collection increases. Evolution (changing characteristics over time, such as acquiring a new source of documents in an unfamiliar format) poses a difficulty for these techniques as well, since they cannot handle documents of new characteristics until a significant number of examples of the new characteristics have been encountered.

In contrast to machine learning approaches, rule-based methods [17], [18] of metadata generation use a set of rules that define how to extract metadata based on human observation. The advantages of such approaches are that they can be implemented in a straightforward manner without training. However, the disadvantages of typical rule-based approaches are their lack of adaptability, difficulty in working with a large number of features, and difficulty in tuning the system since the rules are very rigid. Furthermore, heterogeneity of document characteristics can result in complex rule sets whose creation and testing can be very time-consuming [19]. Complexity of rule-based methods will grow much more than linearly in the number of rules, in which case even a well-trained rule-writer will be hard-pressed to cope with changes in an evolving heterogeneous collection and maintain a conflict-free rule set.

Senbazuru proposed by [20] is a prototype of spreadsheet database management system. Senbazuru allows users to search for relevant spreadsheets in a large dataset, probabilistically creates a relational version of the data, and provides several relational operations over the resulting extracted data. The extract component of Senbazuru consists of a sequence of modules that convert the data in each spreadsheet into the relational model. The first module identifies data frame structures in each spreadsheet. The next module is the hierarchy extractor, which recovers the attribute hierarchies for attribute regions of the data frame. In each attribute region, the module justifies which attributes describe which other attributes. However, this prototype cannot define categories of the spreadsheets and semantic of extracted data which allow users search spreadsheets efficiently.

The proposed approach can be seen as a variant of the rule-based approach. However, the time complexity of semantic metadata, which is induced by heterogeneity and evolution, is effectively reduced by categorizing spreadsheets by layout similarity and by providing flexible metadata extraction rules for each category.

VII. Conclusion and Future Work

This paper has proposed *SEMEXSS* which is a novel rule-based semantic metadata extraction system for spreadsheets. Metadata extraction is based on spreadsheet categories, location relationship between metadata and its identifier of a spreadsheet. The system is applied to spreadsheet collections which are categorized by layout similarity. Metadata extraction starts from the process of selecting a sample spreadsheet from a spreadsheet collection of the same category. System manager defines a XML Schema specifying metadata for the sample spreadsheet. The location of metadata of the sample spreadsheet is defined by mapping schema elements to spreadsheet cells storing metadata. Given the sample spreadsheet, the system generates category justification rules and semantic metadata extraction rules of the spreadsheet collection. This paper has also introduced *MetadataExt* which is an algorithm extracting semantic metadata from given spreadsheet, the metadata schema set, the category justification rule set and the metadata extraction rule set. Experiment result of *MetadataExt* shows that the semantic metadata generation needs amount of time that is directly proportional to multiplication of the number of metadata extraction rules and the number of spreadsheet's cells storing text strings.

This work leaves some space for system extension. *SEMEXSS* is designed to handle spreadsheets of Excel. It should be extended to generate semantic metadata from spreadsheets of other software. Furthermore, metadata generated from captions of figures of spreadsheets should be taken in the account in order to enable the search system to search spreadsheets from metadata of figures of the spreadsheets.

REFERENCES

[1] Microsoft Excel. [Online]. Available: http://office.microsoft.com/en-us/excel/excel-2010-features-and-benefits-HA101806958.aspx, 2010.

[2] IBM. Lotus 1-2-3. (2013). [Online]. Available: http://www-01.ibm.com/software/lotus/ products/ 123/

[3] OpenOffice, Calc: The all-purpose spreadsheet. (2013). [Online]. Available: http://www.openoffice.org/product/calc.html

[4] Copernic, Copernic Desktop Search 3.7. (2013). [Online]. Available: http://www.copernic.com/en/products/desktop-search/index.html

[5] X1 Technologies, X1 Professional Client. (2013). [Online]. Available: http://www.x1.com/products/professional-client

[6] Soft Experience, Metadata Miner Pro. (2012). [Online]. Available: http://peccatte.karefil.com/software/Catalogue/MetadataMiner.htm

[7] Z. Guo and H. Jin, "A rule-based framework of metadata generation from scientific papers," in *Proc. 10th International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, JiangSu, China, 2011, pp. 400-404.

[8] S. Chatvichienchai, "Spreadsheet metadata generation: A layout-based approach," in *Proc. 23rd International Conference of Database and Expert Systems Applications (DEXA)*, Austria, 2012, pp. 147-160.

[9] W3C, *Extensible Markup Language (XML) 1.0*, 4th ed. 2006.

[10] W3C, *XML Schema*, 2001.

[11] Microsoft Excel. (2013). [Online]. Available: http://office.microsoft.com/en-001/training/ways-to-create-xml-maps-RZ001131077.aspx?CTT=1&section=5

[12] J. Walkenbach, *Excel 2010 Power Programming with VBA*, 1st ed. Wiley, 2010.

[13] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox, "Automatic document metadata generation using support vector machines," in *Proc. the 3rd ACM/IEEE–CS Joint Conference on Digital libraries*, 2003, pp. 37-48.

[14] H. Han, E. Manavoglu, H. Zha, K. Tsioutsiouliklis, C. L. Giles, and X. Zhang, "Rule-based word clustering for document metadata generation," in *Proc. SAC, LNCS*, 2005, vol. 3897, pp. 1049-1053.

[15] B. Cui, "Scientific literature metadata generation based on HMM," in *Proc. 6th Int. Conference of Cooperative Design, Visualization, and Engineering 2009*, Luxembourg, 2009, pp. 64-68.

[16] J. Tang, K. Maly, S. Zeil, and M. Zubair, "Automated building of OAI compliant repository from legacy collection," in *Proc. the 10th International Conference on Electronic Publishing*, Bulgaria, 2006, pp. 101-112.

[17] D. Bergmark, "Automatic generation of reference linking information from online documents," *CSTR 2000-1821*, 2000.

[18] S. Mao, J. W. Kim, and G. R. Thoma, "A dynamic feature generation system for automated metadata generation in preservation of digital materials," in *Proc. the First Int. Workshop on Document Image Analysis for Libraries*, Los Alamitos, 2004, vol. 225.

[19] S. Klink, A. Dengel, and T. Kieninger, "Document structure analysis based on layout and textual features," in *Proc. Fourth IAPR International Workshop on Document Analysis Systems*, 2000, pp. 99-111.

[20] Z. Chen, M. Cafarella, J. Chen, D. Prevo, and J. Zhuang, "Senbazuru: A prototype spreadsheet database management system," in *Proc. the VLDB Endowment*, Trento, Italy, 2013, vol. 6, no. 12, pp. 1202-1205.

Somchai Chatvichienchai received the B.S., M.S. degrees in computer engineering from Chulalongkorn University in 1977, Kyushu University in 1989, respectively, and the PhD degree in informatics from Kyoto University in 2004. Somchai joined the Department of Info-media at Siebold University of Nagasaki in 2004 and was a full-time professor of the Dept. of Information and Media Studies in 2009. Dr. Somchai's research interests include database theory and systems, XML, access control, and information retrieval. Dr. Somchai is a member of the ACM, IEEE, IAENG, and the Database Society of Japan (DBSJ).