Towards Automatic Capturing of Semi-structured Process Provenance

Andreas Wombacher, Mohammad Rezwanul Huq

Towards Automatic Capturing of Semi-Structured Process Provenance

Andreas Wombacher and Mohammad Rezwanul Huq

University of Twente, 7500 AE Enschede, The Netherlands
(a.wombacher,m.r.huq)@utwente.nl

Abstract. Often data processing is not implemented by a workflow system or an integration application but is performed manually by humans along the lines of a more or less specified procedure. Collecting provenance information in semi-structured processes can not be automated. Further, manual collection of provenance information is error prone and time consuming. Therefore, we propose to infer provenance information based on the file read and write access of users. The derived provenance information is complete, but has a low precision. Therefore, we propose further to introducing organizational guidelines in order to improve the precision of the inferred provenance information.

1 Introduction

Semi-structured processes are business or scientific workflows, where the execution of the workflow is not completely controlled by a workflow engine, i.e., an implementation of a formal workflow model. Examples can be found in scientific communities where a workflow of data cleansing, data analysis and data publishing is informally described but the workflow is not automated. Other examples can be found in scenarios where several people potentially from different organizations cooperate e.g. in creating a yearly progress report or writing a scientific paper.

The lack of a workflow system controlling the process execution and the usage of non-provenance aware applications for executing activities, i.e., the application is not recording provenance information itself, makes it difficult to answer provenance questions like 'Who did what when and how?'. Since the provenance question can not be answered it is hard to assess the quality of the output of the semi-structured process, like e.g. the quality of data in a report. In many scenarios, manual provenance acquisition is infeasible since it is too time consuming, and therefore too costly. Thus, the aim is to explore automatic provenance capturing in semi-structured processes.

By analyzing the execution of semi-structured processes the following observations can be made: First, the result of an activity is a single document or a set of documents.

Second, information contributing to a document is often copied from previous activity results, i.e., other documents. Thus, for an activity the source document must have been read/opened before the information can be copied and saved in the target document. Therefore, each document which has been read before a save operation is performed may have contributed to the target document.

Third, users of ICT systems organize their information often in hierarchical structures like e.g. directories. Thus, for a certain activity a user has to perform specific directories are relevant.

Fourth, several revisions of documents are issued over time, where the revision number may or may not be explicated in the file name. However, the process handles potentially various revisions of a document.

In this paper, an approach to automatically capture provenance information for semi-structured processes is proposed. The basic idea is that all documents, which have been read by a user, may have contributed to a document saved later on. Since this derived provenance is rather imprecise it is proposed to facilitate knowledge of the process, of the revisions of documents, and of the organization of the documents by the user to increase the precision. In particular, it is proposed to introduce organizational directives, i.e., guidelines for the user on how to organize information relevant to the data processing workflow. The more strict these guidelines are the higher the precision that can be achieved. However, strict guidelines lower the degree of freedom for a user to organize 'her' data and therefore may result in non complying users effecting the data quality.

In a next step we investigate the mining of provenance relations from the collected version information of the used files without considering the organizational directives. It turns out that actions performed by a script or program can be mined well while manual modifications of files can be detected but the provenance can not be mined. The results are qualitatively compared with the provenance information derived from the organizational directives.

The proposed approach is based on a WebDAV infrastructure which supports version control of files. The proposed approach has been implemented and evaluated on the paper writing of this paper.

In the following related work is discussed (Sect 2) before a use case is introduced (Sect 3). The approach is presented on a conceptual level (Sect 4) while a more technical view on the derivation of provenance information is provided in Sect 5. Next an evaluation of the proposed approach (Sect 6) is presented followed by a description of the mining approach (Sect 7) and the conclusions (Sect 8).

2 Related Work

Automatic collection of provenance information is often applied in e-science workflow systems, like e.g. Kepler [1] or Taverna [2]. Most systems even rely on exchanging data via files. In previous work [3] we investigated inference of provenance information for stream processing workflow system using a temporal model. However, the workflow system is executing a workflow and all involved tasks are executed automatically which is a major difference to our requirements.

Provenance information is also collected in closed systems like e.g. a data warehouse [4] or a relational database [5]. The level of granularity in these approaches varies between fine-grained and coarse-grained data provenance [6]. The fact that the provenance acquisition is limited to the system makes it infeasible for our scenario.

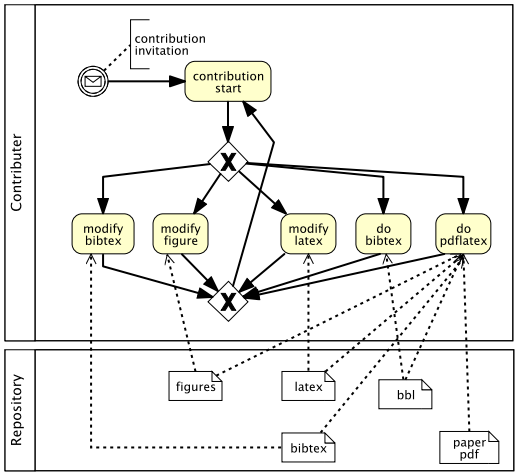

Fig. 1. Editing phase of a paper writing process

Automatic collection of provenance information focusing on the exchange of information is addressed in various ways. In [7] provenance information is captured by monitoring a service bus where the invoked services must not be provenance aware. Although this is quite close to the scenario at hand, in this case all processing has to be implemented as a service, which is not the case in our case. The approach in [8] records file manipulation operations including system variables and their changes. However, this approach is limited to a single computer since it is woven into the kernel of a linux system which differs significantly from our scenario. This may be the closest match to our approach. In [9] the authors automatically collect provenance information based on events recorded by browsers. However, the approach is limited to a single application.

Systems for storing provenance information are e.g. Tupelo2 [10] or Karma2 [11]. These systems provide an into store provenance information and provide means to query the data. The provenance information derived by the proposed approach could be stored in such a system. Further, the acquired provenance information could be made accessible in different provenance models like e.g. the Open Provenance Model (OPM) [12] or the value centric model (TVC) [13]. However these are just alternative representations of the derived provenance data, while the focus of this paper is on acquiring the provenance data rather then how to represent them.

3 Use Case

The process used as a running example is writing this technical paper with multiple authors in Latex. After initializing the project, two authors are writing together on the same paper. A BPMN notation of the process is depicted in Fig 1[1].

In the process the following activities can be executed in arbitrary order with an arbitrary number of repetitions:

- creation, update and conversion activities on figures and graphics files (modify figure)
- creation and update activities on bibtex files (modify bibtex)
- creation and update activities on latex files (modify latex)
- pdflatex activity, for creating a pdf file of the paper (do pdflatex)
- bibtex activity, for creating the bibliography file (bbl file) related to the paper (do bibtex)

All activities result in a single result or output document/file. There are two authors involved in this particular paper writing process, which makes it interesting to determine whether a specific revision of a generated pdf file contains all the latest file revisions. Especially whether all figures have been properly converted before executing the pdflatex task. Further, it can be inferred whether in a specific pdf revision of the paper all indexes and the bibliographic information is up to date, since this requires the following task sequence: pdflatex - bibtex - pdflatex -pdflatex.

4 Approach

The aim is to infer provenance information from a semi-structured process execution without the user providing any information and the legacy applications not being provenance aware. Since files are used to exchange information between different activities of the process, the approach is based on documenting data manipulations on files. Combining file manipulation information with data dependencies of the process allows to infer which revision of which file may have contributed to a revision of a file written by a particular user.

The intuition is that data processing is based on zero or several input files producing one output file. In particular, all files which have been opened before the point in time a file is saved potentially contributed to the creation of the saved file. These derived provenance relations have a very low precision, i.e., are too broad especially when considering the amount of files opened in the coarse of a day at a desktop computer.

Therefore, we propose to facilitate knowledge of the process to derive organizational directives, i.e., rules for the user performing the data processing activities to be able to associate files with specific activities in the process. Thus, they support inference of provenance information. Organizational directives give a responsibility to the user without technically enforcing the directive. Organizational directives are widely implemented in organizations, like e.g. you are not allowed to install software on your company laptop, you are not allowed to download copyright protected material, you are obliged to make backups or to encrypt your hard disc.

[1] Created with `http://oryx-project.org/`

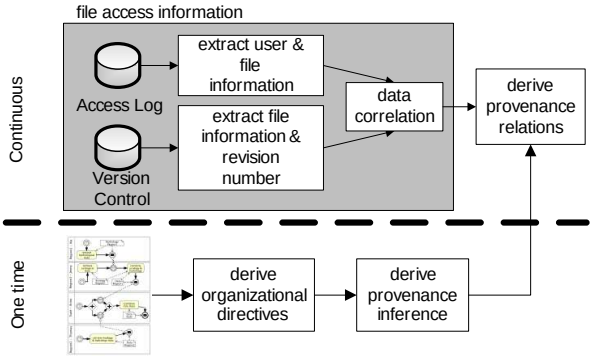

Fig. 2. Prototype data processing

The approach is depicted in Fig 2. Based on the semi-structured process (lower left corner of Fig 2) the organizational directives are derived, of which inference rules for provenance relations can be derived. Further, the derived provenance inference rules are continuously applied on the automatically acquired file access information (gray box in Fig 2) to derive provenance relations. In particular, a relation exists, if a file with revision v_s has been written after a file with revision v_r has been read and $v_r < v_s$ assuming a global version scheme. In the following the basic derivation of provenance relations (continuous part) and the creation of organizational directives (one time part) of the approach are discussed in more detail after the infrastructure and some generic directives are introduced. The inferred provenance information is made accessible via a web application [2].

4.1 Infrastructure to Capture User Behavior

The proposed approach is based on a Web based infrastructure for file storage and version control, in particular, a WebDAV infrastructure facilitating a Subversion (SVN) as version control system. WebDAV is an HTTP based protocol for managing content on a web server, i.e., accessing, writing and moving files. WebDAV seamlessly integrates into various operating systems, i.e., the WebDAV server appears as a mounted network drive and therefore is intuitively usable also for basic ICT users.

Further, since the WebDAV protocol is based on HTTP it easily supports distributed and cross-organizational scenarios, like the motivating example in Sect 3. Furthermore, WebDAV can also be used for information dissemination. From a provenance capturing point of view, WebDAV has the advantage that it supports logging of file read operations via the web server access log and logging of write operations via the underlying version control system. In particular, the web server access log documents the read (GET), write (PUT), move (MOVE), and delete (DELETE) operations applied to files on the server. Further, the Subversion (SVN) version control log documents write (Add, Modify), delete (Delete) and move (combined Add and Delete) operations. Both sources (see

[2] accessible at `http://www.sensordatalab.org/offline_provenance_web/`

content gray box in Fig 2) are required to capture the file information and therefore have to be correlated (see Sect 5.1).

The choice of using WebDAV seems a bit outdated compared to currently existing offers like DropBox, Google Drive or Microsoft SkyDrive. While all these offers provide history information comparable to the SVN functionality, the size of the history information maintained is limited. The history information documents the write access to files, however, the read access to files is not documented there. Furthermore, the products synchronize the files with all related devices instantly, thus, every read operation by another person after a change is a local file operation, which is not documented. Therefore, WebDAV although old fashioned allows to track read access since the WebDAV functionality is under our own control.

4.2 Generic Organizational Directives

The proposed approach is based on observing the handling of files. Thus, some generic directives are needed to ensure that the handling of files can be observed at the first place. Similar to directives in organizations that all important information has to be saved on a network drive because local disks are not backuped, a data security directive is introduced requiring the user to save data on a mounted network drive, i.e., a WebDAV server.

Directive 1 (Data Security) *Users must save all files related to a semi-structured process on the network drive. Thus, it is not allowed to store files related to the process on a local disc.*

This directive is necessary, since local file systems can not be monitored with regard to file handling that easily from outside the computer.

A standard directive in organizations is that login and password information must not be shared between different users. It must always be possible to identify a responsible user for any observed action in the infrastructure.

Directive 2 (Delegation) *In case of vacation or illness the execution of activities must be delegated.*

In many organizations you have a clean desk policy, which means that at the end of the day all business relevant documents must be removed from the desk. Translating this into the digital world means that the desktop computer has to be switched off at the end of the day, which is also in the context of green IT getting more attention. This results in the following directive:

Directive 3 (Clean Desk Policy) *The user must shut down his/her computer at the end of the day, i.e., there are no open files or applications active anymore.*

Provenance relations are based on the observed reading and saving of files. Since it can not be observed by WebDAV when a file is closed again, the clean desk policy directive enforces that at the end of a day all files are closed. Thus, this is a synchronization point for deriving provenance relations by excluding files opened at previous days.

4.3 Provenance Relations

The generic directives are the basis to infer provenance relations based on WebDAV commands. A provenance relation is a relation between a read or save operation of a file A and a save operation of a file B, where the read or save operation of file A is performed before the save operation of file B. The order of the operations can be determined by the timestamps at which the operations are observed and on the associated revision numbers.

Due to directive 3 only read operations which have been performed at the same day as the write operation are considered. Further, according to directive 2 each file access is associated with a specific user, thus, provenance relations require that the files are read and saved by the same user.

Based on the scenario described in Sect 3 it can be derived that e.g. the modification of a latex file will not use information from an image file. This independence, which is derivable from the process, has to be explicated as specific organizational directives as discussed in the following section.

4.4 Specific Organizational Directives

The goal of the organizational directives is to increase the precision of data provenance by excluding observed provenance relations which are actually irrelevant. Since provenance relations are based on sets of read files with a single point in time when files are certainly closed again (see Directive 3), additional mechanisms are required to determine the precise inference of data provenance. One way is to apply directives on the hierarchical structure of data on the network drive. Further such a structure is often established to exchange information between different organizations and to control access rights. In particular, one possible approach is to establish a basic directory structure where each directory is associated to a particular activity in a data processing workflow, as it is known e.g. from group ware solutions like BSCW or groove. An alternative is to use filters based on regular expressions exploiting file naming conventions or specific file extensions being unique for the output of an activity.

However, the derived directives might be not specific enough or too complicated to implement resulting in imprecise provenance information. The challenge is to find a good balance between usability of organizational directives and the targeted data provenance precision.

With regard to the motivating scenario in Sect 3 the following directives should be instantiated:

Directive 4 (Scientific paper writing)
1. *All bibtex files are located in a bib subdirectory of the project root directory.*
2. *All pictures and figures are located in a pics subdirectory of the project root directory.*
3. *All latex files, project files and auxiliary files are located in the project root directory.*
4. *Figures may require a conversion of file formats. The source figure filename is a prefix of the target figure filename ignoring file extensions.*

5. *An update of a bibtex file only depends on the old bibtex file, i.e. the filename of source and target are equivalent.*
6. *The execution of a pdflatex command reads data from all project directories and writes a pdf file in the project root directory.*
7. *The execution of a bibtex command reads from the project root directory and the bib directory and writes a bbl file in the project root directory.*

5 Provenance Relation Derivation

After the conceptual introduction in the previous section a more detailed technical discussion follows. The provenance information forms a provenance graph, which consists of vertices and edges. A vertex is either an access log entry or a SVN log entry. The edges represent the provenance relations. In this paper three classes of provenance relations are distinguished: provenance relation correlating SVN and access log entries, SVN step relations, and relations derived from directives. A access log contains read (GET), write (PUT), move (MOVE), and delete (DELETE) operations (WebDAV commands) applied to files. Further, the SVN internal logs contains write (Add, Modify), delete (Delete) and move (combined Add and Delete) entries.

All relation classes are discussed in the following subsections after a discussion on observations of file handling using WebDAV clients.

5.1 File Handling Observations

The manipulation of files via a WebDAV client and documenting them in SVN and access logs is not as straight forward as initially expected. In particular, the following observations can be made:

First, adding and updating files is realized in different combinations of WebDAV commands by different applications.

Second, the WebDAV MOVE command documents only the source filename of the move in the access log, but not the destination filename. However, by using WebDAV with autocommit each change on a file results in a new revision number, thus, only a MOVE command results in a delete and an add SVN log entry with the same revision number. This can be facilitated to correlate SVN and access log entries.

Third, a WebDAV DELETE command removes a file from the SVN repository. After the command is executed the file is not in the repository anymore, thus, the revision number of the removal of the file requires an extra query to the SVN log.

Fourth, reading a file is only documented in the access log. Which revision of a file has been read has to be inferred from the state of the SVN and the size of the file read.

The final observation is that the correlation between SVN and access log entries can not be based on time, since the timestamps recorded by the SVN and the access log are points in time when the event has been recorded in the corresponding system. Thus, the timestamp of the access log entry is always before the entry in the SVN log since the HTTP request is first processed by the HTTP server which forwards the request to the WebDAV and therefore to the SVN. However, in case of two fast subsequent operations,

it is possible that the access log has two entries before any entry is recorded in the SVN log. Thus, it is sometimes hard to infer the correlation of access to SVN log entries.

Based on these observations, a simplified version of the correlation algorithm is depicted in Alg 1. The token variable indicates whether the SVN entry can be correlated to an access log entry. A correlation is possible if the SVN entry is either a Delete or an Add entry (line 3). Further, a correlation is possible if the SVN entry is a Modify entry and the previous entry has been older than 2 seconds (line 4). The two second bound is based on the observation that an incremental upload of a file to the SVN resulted in a new revision approximately every second in our test system.

If an entry can not be correlated to an access log entry (line 7), than it is inferred that an incremental upload is occurring. Incremental updates are documented as a provenance relation between SVN log entries called *SVN step* relation (line 9). Otherwise, Add and Modify entries are correlated with PUT entries (line 11) and Delete entries are correlated either with DELETE (line 12) or with MOVE entries (line 13).

Since the file access is not documented in the SVN log entries, a second loop is executed on the access log entries (line 14). In particular, all GET entries are selected (line 15). For each GET entry the corresponding SVN entry is inferred (line 16) and the entries are correlated (line 17). In Fig 3 an example of the access and SVN log are

```
1  token=true;
2  forall the SVN entries do
3      if Delete or Add then token=true;
4      if Modify and size>0 and time difference to previous event<2sec then
5          token=true;
6      if not token then
7          ignore entry for correlation;
8          document SVN step relation with previous event;
9      else
10         if Add or Modify then correlate to PUT;
11         if Delete then
12             correlate to DELETE;
13             if not possible then correlate to MOVE;
14 forall the access log entries do
15     if GET then
16         find SVN entry preceding the GET with same filename and size;
17         correlate entries;
```

Algorithm 1: Simplified Data Fusion

depicted. Saving file *pics/prototype.vsd* (F1) results in a PUT entry in the access log and two entries in the SVN log: an Add entry and a Modify entry. The Add and Modify entry in the SVN log are related by a SVN step relation. Further, the PUT access log entry is correlated with the Add SVN log entry. Saving file *pics/prototype_architecture.eps* (F2) results in a single entry in the access and SVN log. Reading the file F1 results in a GET

access log entry which is correlated to the last write operation of file F1 in the SVN log, i.e., the Modify entry resulting on version 2.

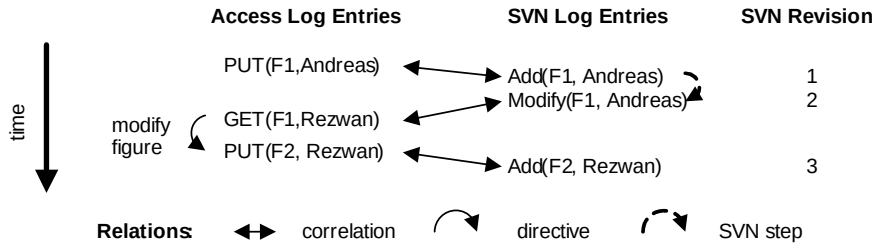

Fig. 3. Derived relations for the paper writing use case

5.2 Directive Provenance Relations

The last type of relations are the ones derived from organizational directives. This type of relation is inferred between access log entries only. As outlined in Sect 4 all files read at the same day as another file is saved are potentially contributing to the saved file. The relation is labeled by the associated activity in the process. The organizational directives can be translated into constraints on these potential provenance relations mainly by constraining files locations and filenames of sources and targets of a provenance relation.

For example Directives 4 in Sect 4.4 determine that only files in directory *pics* are relevant for the 'modify figure' activity (see Fig 1). Thus, files of other directories do not contribute to any provenance relation associated to this activity.

In Fig 3 an example of the 'modify figure' directive relation is depicted. Since file *pics/prototype.vsd* (F1) is a prefix (ignoring the extension) of file *pics/prototype_architecture.eps* (F2) as specified in directive 4.4 a directive relation between the *GET(F1, Rezwan)* access log entry and the *PUT(F2, Rezwan)* access log entry can be inferred.

6 Evaluation

To evaluate the proposed approach the running example described in Sect 3 is used. The scenario is a semi-structured process of writing a scientific paper using latex, where figures are usually created in Microsoft Visio, which are then converted to Enhanced Windows Metafile (emf) files, and further into Encapsulated PostScript (eps) files. The pdflatex command uses a library to convert the eps figure files into pdf figure files, which can then be used in the resulting pdf file.

The process uses the generic organizational directives (directives 1-3 in Sect 4.2) as well as the specific organizational directives (directives 4.1-7) discussed in Sect 4.4.

These organizational directives are not hard to implement, since we use the same way of structuring a paper writing project for years already. Our guess is that for many

semi-structured processes this is similar, since people tend to organize their data rather on content and topics than on time.

6.1 Overhead

The paper is based on two latex files, which uses three style files contained in WebDAV. There are six bibtex files used and one image file with its corresponding emf, eps and pdf files. Further there is one BPMN file directly available as pdf file. The processing uses 4 auxiliary files and writes a single pdf file.

The overhead for the user during this experiment is clearly experienced due to the network delay introduced by using WebDAV compared to a local disc. My ADSL line at home is a bout a 1000 times slower than my local disc access. In particular, for a pdflatex command 840kB are read and 377kB are written, thus, in total 1.2MB are transferred. Further, for a bibtex command 250kB are read and 3kB are written, thus, in total 253MB are transferred. So far we have approximately 120 executions of the pdflatex command during the complete paper writing process. As a conclusion, since a build is not performed that often, the overhead is effecting our working experience only marginally.

The usage of WebDAV has an influence on the network load. How much depends on the size of the files the users are working with. In the context of latex these are rather small files and therefore the effects are hardly measurable. In case of other applications using e.g. spatial information or large image files with many write operations this may be different.

6.2 Quality

Quality assessment is problematic since there is no ground truth. Capturing the ground truth automatically is difficult. It would require to extend the operating system and monitor all content related actions, such as opening and closing files, copy and past operations from one application/file to another etc. This includes also the handling of content by non UI applications like e.g. running a pdflatex command. In this case file read system level routines have to be monitored. With the current available resources this is not feasible. In the provenance community there are groups of people researching on how to do this best, however, we are not aware of a complete solution yet - especially not for windows systems.

As a consequence the only possibility is to manually assess the quality of the inferred provenance information by inspection. The manual inspection does not show any missing provenance relations. Sometimes additional provenance relations are reported, which are e.g. artifacts of file transfers (temporary files). From our inspection we have not found cases with missing information. The precision depends on the organizational directives, which can be adjusted to the required level of precision as discussed before. Please be aware that the current alternative is no provenance information at all.

7 Mining Provenance Patterns

The mining approach uses the available log information and applies some counting and some statistic methods to infer provenance relations between different filenames. In particular, the assumption is if an activity is performed repeatedly then the same relations must show up in the log files. In a first step the time difference between file accesses is used to mine provenance relations and in a second step the frequency of file updates is considered.

7.1 Time based Mining

Before the actual mining can start the available dataset has to be investigated with regards to the characteristics of accessing files. The time behavior of a system depends besides others on the used application, the system, and the network connection. Therefore it is not possible to provide an absolute number here.

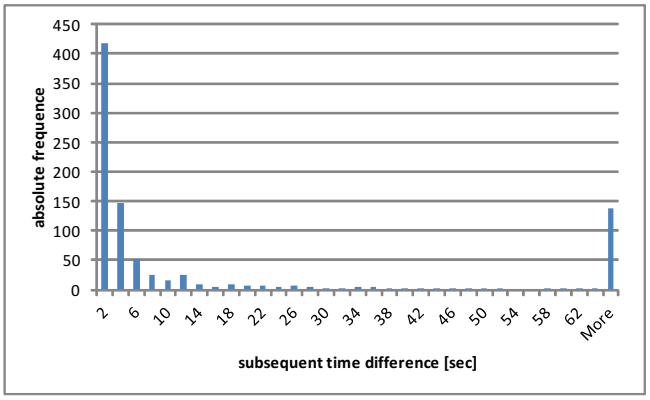

Fig. 4. Time difference in seconds between subsequent entries in the log file

Thus, a histogram of the time difference between two subsequent entries in the log is calculated (see Fig 4). In the histogram you can see that many files are accessed within a time difference of two seconds. The histogram turns flat for time differences greater than 12 seconds therefore, in the following we use the 12 seconds as an upper bound to infer provenance relations between files.

Next we retrieve all pairs of log entries which are less than 12 seconds apart and appear at least twice, remove transitive relations, and represent these relations as a weighted graph, where the weight corresponds to the number of observed occurrences of the relation. To enforce a directed weighted graph, the directionality is enforced by considering only the direction with the higher weight. The directed graphs are depicted in Fig 5 and 6, where the thickness of the connection corresponds to the weight of the relation.

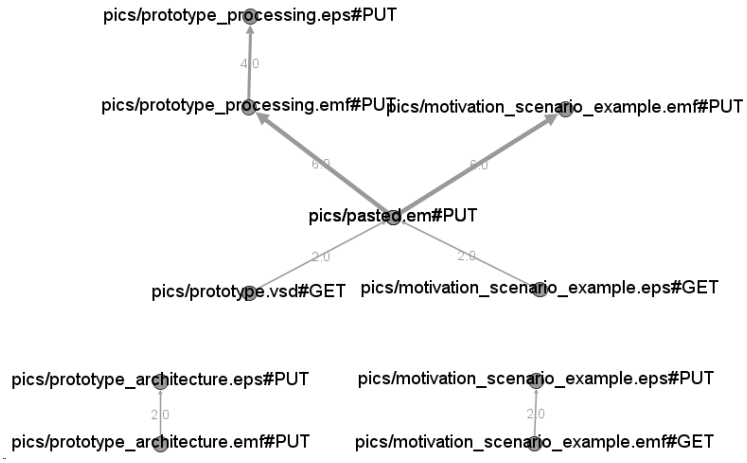

Fig. 5. Cluster 1

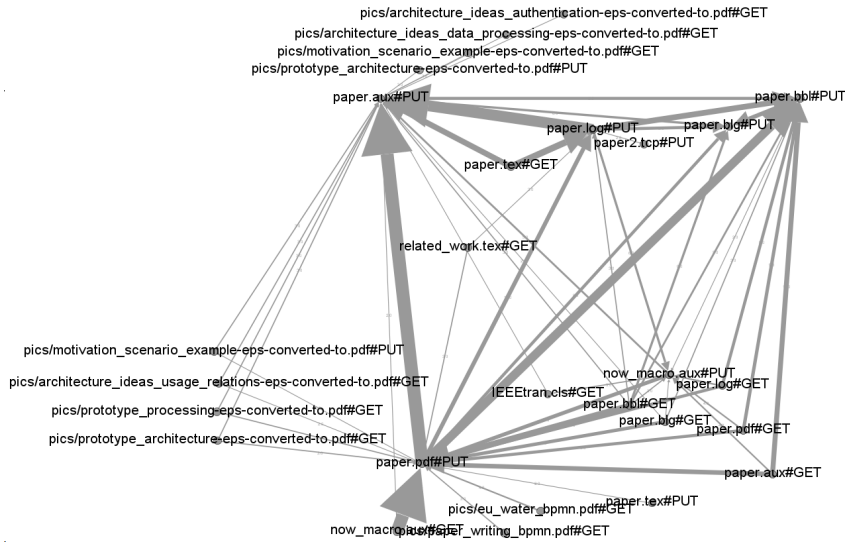

Fig. 6. Cluster 2

The relations result actually in four unconnected graphs. The graph in Fig 6 corresponds to the provenance related to executing the *do pdflatex* activity and the *do bibtex* activity of the BPMN model (see Fig 1). The two activities are captured in one graph since the underlying dataset was produced by using a tool, which provides a macro to perform the sequence *pdflatex-pdflatex-bibtex-pdflatex* automatically. Therefore no sufficient big time difference can be observed between the provenance related to the *bibtex* and the *pdflatex* activity. This graph also contains the conversion of figures from eps

format to pdf format which is performed by a script related to the *pdflatex* command. This is visible by the writing of pdf files with the suffix *eps_converted-to*. The corresponding eps files are not included since the weights of the relations must be at least two.

An unexpected node in this graph is the writing of file *paper.tex* since it is not related to performing the two activities. The explanation for this is that the used tool before executing the above sequence of *bibtex* and *pdflatex* commands checks whether all source files have been saved. If this is not the case then the tool automatically saves the modified files. Therefore, the writing of the latex source file *paper.tex* becomes a part of this graph.

The graphs in Fig 5 are related to figure conversion. It contains the conversion of figures from the emf format into eps format initiated by the author using a conversion tool. The conversion of the other figures is not represented in this graph since a frequency of at least two relations was required. The graph containing the visio file (vsd extension) represents the copying of a visio figure into an emf file.

Overall it can be concluded that all graphs are related to some automated processing initiated by the user. However, manual modifications of files are not observed since they usually require more than 12 seconds to perform the modification, like e.g. writing a new section in the paper.

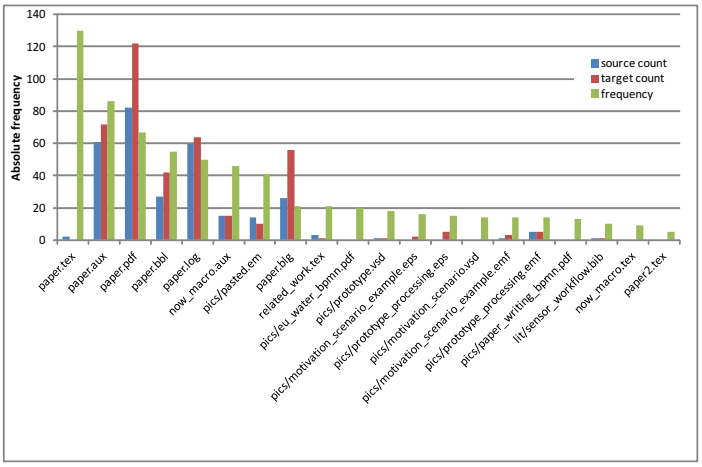

Fig. 7. Histogram of occurrence of files as a source or as a target in an identified provenance relation compared to the frequency of the file being written (incomplete)

7.2 Frequency Analysis

Next we try to identify the files which are mainly manipulated manually. To do so we calculate the histogram of files being written as partially depicted in Fig 7. This histogram is compared with the histogram of the files being part of the provenance

relations as discussed in the previous subsection. A file can be a source or the target of the provenance relation. In Fig 7 the two ways of participation are explicated per file. If a file is written often and participates a lot in provenance relations like e.g. file *paper.pdf*, than it is unlikely that this file is subject to manual manipulation. On the other hand side if a file hardly participates in a provenance relation but is written often then it is most likely manipulated manually often like e.g. file *paper.tex*. Files at the tail of the histogram are potentially subject to manual modification.

Since it is not possible to mine the sources used for manual manipulations of a file without looking at the content of the files, the proposed approach is to ask the user to provide patterns on how to derive provenance relations for these files. In case of figure and latex files the provenance relation mainly includes the other figure and latex files often with the same name, just different versions. The same applies for bibtex files.

The decision on how much of the tail of the histogram is considered determines the imprecision of the derived provenance relations compared to the manual effort required to explicate these rules.

7.3 Evaluation

Comparing the clusters and the manual manipulations identified in the previous two sections and compare them to the organizational guidelines specified in Sect 4.4 shows some interesting results: The mining based approach does not make any assumptions on the behavior of the user. However, in case of manually modified files user input is required to derive the provenance relations. The files to which this applies can be mined automatically.

The organizational directions distinguish the *pdflatex* and *bibtex* activity. However, the mining approach identified additional automated activities like the conversion of emf to eps files and the conversion from eps to pdf files. This indicates that the organizational guidelines may be incomplete.

Comparing the mined graph for *bibtex* and *pdflatex* activity with the provenance derived using organizational guidelines it turns out that the mined approach covers pretty well the real provenance although it may have some additional provenance relations. This is because the processing of *bibtex* and *pdflatex* command in a sequence by the tool results in mined provenance relations which are actually not existent.

8 Conclusion

In this paper, a provenance capturing approach for semi-structured processes involving provenance unaware legacy systems is proposed. Further, the derivation of different classes of provenance relations is discussed. It has been argued that the recall of the proposed approach is very high while the precision depends on used organizational directives, i.e., constraints on handling files as a basis for deriving provenance relations.

Future work should address the currently used 'version on every write' approach to minimize the versions used. A further topic is instead of deriving organizational directives to observe directives and assess the quality of the data handling as applied by the users for deriving provenance relations.

References

1. Ludascher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific workflow management and the Kepler system. Concurrency and Computation: Practice and Experience **18**(10) (2006) 1039–1065
2. Oinn, T., Addis, M., Ferris, J., Marvin, D., Greenwood, M., Carver, T., Pocock, M., Wipat, A., Li, P.: Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics **20**(17) (June 2004) 3045–3054
3. Huq, M.R., Wombacher, A., Apers, P.M.G.: Facilitating fine grained data provenance using temporal data model. In: Proc 7. Intl Workshop on Data Management for Sensor Networks, DMSN, ACM (September 2010) 8–13
4. Cui, Y., Widom, J.: Lineage tracing for general data warehouse transformations. VLDB Journal **12**(1) (May 2003) 41–58
5. Szomszor, M., Moreau, L.: Recording and reasoning over data provenance in web and grid services. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. (2003) 603–620
6. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Rec. **34**(3) (2005) 31–36
7. Allen, M.D., Chapman, A., Blaustein, B.T., Seligman, L.: Capturing provenance in the wild. In: IPAW. Volume 6378 of LNCS., Springer (2010) 98–101
8. Seltzer, M., Muniswamy-Reddy, K.K., Holland, D.A., Braun, U., Ledlie, J.: Provenance-aware storage systems. In: Proceedings of the USENIX Annual Technical Conference (USENIX'06). (June 2006)
9. Margo, D.W., Seltzer, M.I.: The case for browser provenance. In Cheney, J., ed.: Workshop on the Theory and Practice of Provenance, USENIX (2009)
10. Futrelle, J.: Tupelo server http://tupeloproject.ncsa.uiuc.edu/.
11. Simmhan, Y.L., Plale, B., Gannon, D.: Karma2: Provenance management for data driven workflows. Intl J of Web Services Research **5** (2008) 1–23
12. Moreau, L., Freire, J., Futrelle, J., McGrath, R., Myers, J., Paulson, P.: The open provenance model: An overview. Provenance and Annotation of Data and Processes (2008) 323–326
13. Misra, A., Blount, M., Kementsietsidis, A., Sow, D., Wang, M.: Advances and Challenges for Scalable Provenance in Stream Processing Systems. Provenance and Annotation of Data and Processes (2008) 253–265