Groupware Mail Messages Analysis for Mining Collaborative Processes (poster paper)

Claudio Di Ciccio[1], Massimo Mecella[1]
Monica Scannapieco[2], and Diego Zardetto[2]

[1] Sapienza – Università di Roma
Dipartimento di Informatica e Sistemistica Antonio Ruberti
Via Ariosto 25, Roma, Italy
{cdc,mecella}@dis.uniroma1.it
[2] Istituto Nazionale di Statistica
Via Balbo 16, Roma, Italy
{scannapi,zardetto}@istat.it

Nowadays, the most of the research related to workflows has considered the management of formal business processes. There has been some discussion of informal processes, often under names such as "artful business processes", e.g., [1]: informal processes are typically carried out by those people whose work is mental rather than physical (managers, professors, researchers, etc.), the so called "knowledge workers" [2]. With their skills, experience and knowledge, they are used to perform difficult tasks, which require complex, rapid decisions among multiple possible strategies, in order to fulfill specific goals. In contrast to business processes that are formal and standardized, often informal processes are not even written down, let alone defined formally, and can vary from person to person even when those involved are pursuing the same objective. Knowledge workers create informal processes "on the fly" to cope with many of the situations that arise in their daily work. While informal processes are frequently repeated, since they are not written down, they are not exactly reproducible, even by their originators, nor can they be easily shared. Their outcome releases and their information exchanges are very often done by means of e-mail conversations, which are a fast, reliable, permanent way of keeping track of the activities that they fulfill.

The objective of the research proposed in this position document is to automatically build, on top of a collection of e-mails, a set of workflow models that represent the artful processes which lay behind the knowledge workers activities.

A company can take many advantages out of this work. First of all, the unspecified agile processes that are autonomously used become formalized: since such models are not defined *a priori* by experts but rather inferred from real-life scenarios that actually took place, they are guaranteed to respect the true executions and not reflect the expected ones (often Business Process Management tools are used to show the discrepancy between the theoretical workflows and the concrete ones). Secondly, such models can be shared and compared, so that the best practices might be put in evidence from the community of knowledge workers, to the whole business benefit. Moreover, without any further computational cost, an analysis over such processes can be done, so that bottlenecks

Pre-print copy of the manuscript published in the proceedings of SEBD

and delays in actual executions can be found out. We remark here that all of these utilities come out with almost no effort for workers, due to the automated nature of the envisioned approach.

The approach we would like to pursue, in order to retrieve a collection of process models out of an initial set of e-mail messages, involves many research fields at a time, each concerning a sequential phase of the overall processing. For first, we exploit *text categorization* techniques to filter the set of e-mails of interest out of the whole provided collection. Then, we make use of *object matching* algorithms to obtain clusters of related e-mail conversations, from the previous extracted subset. Every cluster is subsequently treated by *text mining information extraction* procedures, in order to find out which tasks e-mail messages are about. Finally, *process mining* is used to abstract process models representing the workflows, which the sets of subsumed tasks were considered traces of. We aim in the future to release a prototype realization of our approach, named MailOfMine.

Background and state of the art. *Process Mining*, also referred to as *Workflow Mining* (see [3]), is the set of techniques that allow the extraction of structured process descriptions, stemming from a set of recorded real executions. Such executions are intended to be stored in so called *event log*s, i.e., textual representations of a temporarily ordered linear sequence of tasks. There, each recorded *event* reports the execution of a *task* (i.e., a well-defined step in the workflow) in a *case* (i.e., a workflow instance). Beware that events are always recorded sequentially, even though tasks could be executed in parallel: it is up to the algorithm to infer the actual structure of the workflow that they are traces of, identifying the causal dependencies between tasks (*condition*s).

The idea to apply process mining in the context of workflow management systems was introduced in [4]. There, processes were modeled as directed graphs where vertices represented individual activities and edges stood for dependencies between them. Cook and Wolf, at the same time, investigated similar issues in the context of software engineering processes. In [5] they described three methods for process discovery: one using neural networks, another with a purely algorithmic technique, the last adopting a Markovian approach. The authors consider the latter two the most promising approaches. The purely algorithmic approach builds a finite state machine where states are fused if their futures (in terms of possible behavior in the next k steps) are identical. The Markovian approach uses a mixture of algorithmic and statistical methods and is able to deal with noise. Note that the results presented in [5] are limited to sequential behavior.

Most of the nowadays mainstream process mining tools model processes as Workflow Nets (WFNs – see [6]), explicitly designed to represent the control-flow dimension of a workflow. From [4] onwards, many techniques have been proposed, in order to address specific issues: pure algorithmic (e.g., α algorithm, drawn in [7] and its evolution [8], α^{++}), heuristic (e.g., [9]), genetic (e.g., [10]). Indeed, heuristic and genetic algorithms have been introduced to cope with noise, that the pure algorithmic techniques were not able to manage.

A very smart extension to the previous research work has been recently achieved by the two-steps algorithm proposed in [11]. Differently from previous works, which typically provide a single process mining step, it splits the computation into two phases: the first builds a Transition System (TS) that represents the process behavior and the tasks causal dependencies; the second makes use of the state-based "theory of regions" ([12], [13], [14]) to construct a WFN which is bisimilar to the TS. The first phase is made "tunable", so that it can be either more strictly adhering or more permissive to the analyzed log traces behavior, i.e., the expert can decide a balance between "overfitting" and "underfitting". Recall, indeed, that event logs are not the whole universe of possible traces that may run: hence, on one hand, the extracted process model should be valid for future unpredictable cases; on the other hand, it should be checked whether such a process model actually adheres to the behavior that most of the gathered traces reflected in the past (we say "most" here to emphasize that a little percentage of the whole log may represent erroneous deviations from the natural flow of tasks). The second phase has no parameter to set, since its only aim is synthesizing the TS into an equivalent WFN. Thus, it is fixed, while the former step can be realized exploiting one among many of the previously proposed "one-step" algorithms (for instance, [9] seems to integrate well).

[2] describes the "ACTIVE" EU collaborative project, coordinated by British Telecom, currently ongoing (due date is February 2011). Such project addresses the need for greater knowledge worker productivity by providing more effective and efficient tools. Among the main objectives, it aims at helping users to share and reuse informal processes, even by learning those processes from the user's behavior.

Object Matching (OM) is the problem of identifying pairs of data objects coming from different sources and representing the same real world object. The aim of OM techniques is typically related to the cleaning of large data sets from erroneous duplicates. Such duplicates usually derive from misspellings, abbreviations, lack of standard formats, or any combination of these factors. Analogous techniques can be used to achieve a *reference reconciliation* (see [15]), namely the identification of related references in complex information spaces where data corresponding to the same reference can be structurally heterogeneous (e.g., e-mail contacts, documents, spreadsheets). If data objects are records, the problem is known in literature as Record Linkage (RL) [16]. Integration of different data sources and improvement of the quality of single sources are only some of the real application scenarios that need to solve the OM problem. The RL version of the OM problem has received considerable attention by the scientific community. Most of the works (e.g. [17], [18]) focus on solving the problem within a relational DBMS: often, in the real world, entities have two or more representations in databases. Duplicate records do not share a common key and/or they contain errors that make duplicate matching a difficult task.

Some techniques for discovering strong entity associations in semi-structured data, such as document meta-data, are also known (e.g., see [19]). Here, the attention is moved towards a novel technique, proposed in [20]. Such a method-

ology differs from the others in the field, indeed, since *(i)* it is able to treat not only records but every kind of objects, provided that it is possible to define a distance for them, *(ii)* it is focused on effectiveness rather than on the ability to manage huge amount of data, *(iii)* its nature is completely automated. Regarding the third point, there were new unsupervised techniques (such as [17]) already proposed, but none of them, to the best of our knowledge, were fully automated. Indeed, though not requiring exactly a clerically prepared training set, such techniques still depend critically on some external inputs (e.g., human intervention is needed to set crucial parameters for the algorithms in [17]). It achieves such a result by applying a two-phases algorithm. It makes use of statistical models that allow to represent a probability distribution as a convex combination of two distinct probability distributions: the one stemming from the sub-population of Matches ($\mathcal{M}$) and the other from that of Unmatches ($\mathcal{U}$). Thus, the two consecutive tasks are: *(i)* estimating mixture parameters by fitting the model to the observed distance measures between pairs; *(ii)* then, obtaining a probabilistic clustering of the pairs into Matches and Unmatches, by exploiting the fitted model. In the clustering step the fitted mixture model is used to search an optimal classification rule such that each pair can be assigned, based on its observed distance value, either to the $\mathcal{M}$ or to the $\mathcal{U}$ class. Such this constrained optimization problem is solved by means of a purposefully designed evolutionary algorithm [21].

Text Mining, or Knowledge Discovery from Text (KDT) deals with the machine supported analysis of text: indeed, it refers generally to the process of extracting interesting information and knowledge from unstructured text. It is a field in the intersection of related areas such as information retrieval, machine learning, statistics, computational linguistics and, especially, data mining.

Natural language text contains much information that is not directly suitable for automatic analysis by a computer. However, computers can be used to sift through large amounts of text and extract useful information from single words, phrases or passages. Therefore, text mining can be interpreted in the sense of an information extraction activity, i.e., as a restricted form of full natural language understanding, where we know in advance what kind of semantic information we are looking for.

Text mining covers many other topics that are out of the scope of this paper: for a comprehensive survey on it, please refer to [22] or, for a more extended explanation, [23]. A particular discipline of interest that belongs to it is *Text Categorization* (TC, also known as *Text Classification* or *Topic Spotting*), namely, the activity of assigning *categories* (symbolic labels), from a given set, to natural language texts, on the basis of *endogenous* knowledge only (i.e., knowledge is extracted from the documents only and not from other possible external sources). The categories in the given set can be two (*Binary* TC, i.e., a document can belong to a category or its complement) or more. TC is applied in many contexts, such as document filtering and automated metadata generation. For a comprehensive survey on Machine Learning in Automated Text Categorization, the reader can refer to [24].

As a successful example of application, we want to report here a case that deeply relates with the research proposal of this paper: [25] proposes a method employing text mining techniques to analyze e-mails collected at a customer center. The method uses two kinds of domain-dependent knowledge: one is a key concept dictionary manually provided by human experts and the other is a concept relation dictionary automatically acquired by a fuzzy inductive learning algorithm. Based upon the work exposed in [26], the depicted method takes as input the subject and the body of an e-mail, decides a text class for the e-mail, extracts key concepts from e-mails and finally presents their statistical information as well.

The MailOfMine proposed approach. As depicted in Figure 1, MailOfMine is intended to be divided into layers that act in series like in a waterfall model. Each layer is built to achieve a specific task (e-mail reading, e-mail filtering, e-mail clustering, tasks extraction, workflow mining). The initial input is a file storing a whole repository of e-mail messages. The final output is a set of process models inferred from the given input. In the middle, every layer is drawn to receive as input the result that comes from the upper one, starting from the raw e-mails, and in turn take the output to the lower one, down to the final process models.

The very first task is accomplished by a plug-in based system, that must be able to extract a common format for e-mail messages, stemmed from heterogeneous sources: for example, Microsoft Outlook, Mozilla Thunderbird, Evolution Mail files use different storage formats, but the system must be able to manage them all.

In Figure 1, such task corresponds to the layer 0 (*Multiple E-Mail Storage Formats Extraction*).

From that point on, the proposed approach follows this pattern, founded on a double analysis/synthesis passage:

1. (*Mail Filtering*) *analysis* on the set of heterogeneous e-mail messages to filter irrelevant messages out (Text Categorization, Information Retrieval);
2. (*Mail Clustering*) *synthesis* on the set of relevant messages, to cluster related ones into homogeneous[3] sets (Object Matching);
3. (*Tasks Inference*) *analysis* on each cluster, to extract the tasks out (Information Extraction);
4. (*Workflows Inference*) *synthesis* on tasks, to build a process model that conforms with the subsumed trace (Process Mining).

Conclusions. In this paper, the MailOfMine approach and its basic idea of inferring artful business process models, i.e., agile workflows formal representations, from knowledge workers e-mail storage files, have been proposed. The

[3] Here "homogeneous" has to be intended with respect to the activities to serve for the purpose of task inference.

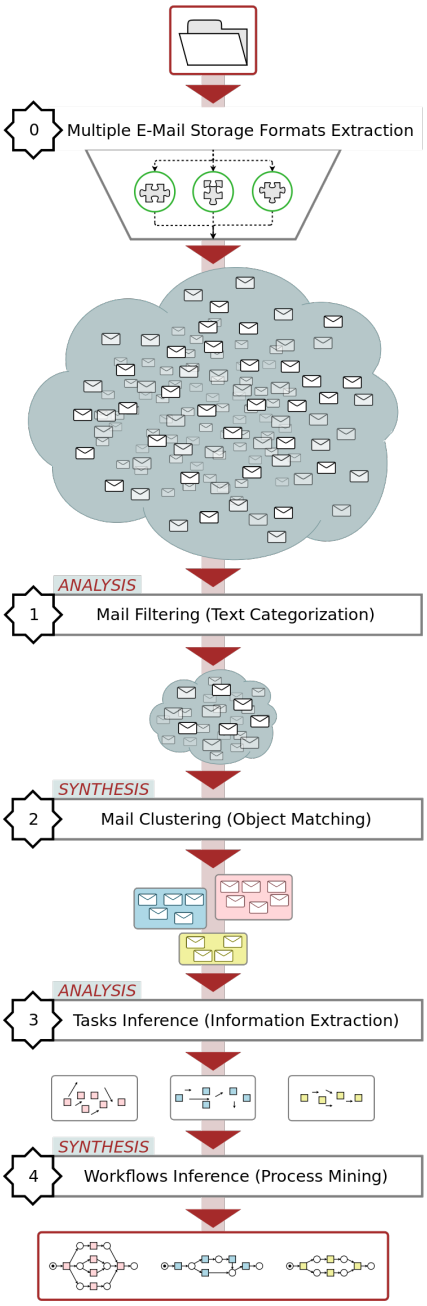

Fig. 1. The MailOfMine architecture

approach can take advantage of many previous works that succeeded in the heterogeneous research fields that are involved in this context (Text Mining, Object Matching, Process Mining). However, new research is indeed needed: the challenge is not only in the integration of various techniques, but also in applying process mining techniques on top of more traditional text mining ones. Currently we are studying the details of all the different layers/steps, and then we will proceed to an effective validation, by applying the approach and the prototype tool to a corpus of about 10 GByte emails, collected over 10 year of work of some authors to Italian and European research projects.

In future work, we want also to address other interesting issues. A challenge is the one of cooperative activities: they may involve many knowledge workers at a time, and it can happen that a task, say *DoIt*, that Mr A. Bloggs demands to Mrs B. Doe, is in turn redirected to Mr C. Smith. Thus, Mr C. Smith fulfills a series of tasks, in collaboration with Mrs B. Doe, which Mr A. Bloggs is unaware of. It could be interesting, then, to investigate on how to relate these activities that are traced by separated e-mail sets (one belonging to Mr A. Bloggs, the other to Mrs B. Doe), so to unify them under a single workflow model.

Another point to cope with is the question of privacy: e-mail messages may contain sensible private information that the single knowledge worker, or the company, might not want to be processed. At this initial stage of our research, we are supposing that treated data are public at least to the company that the knowledge worker works for, since we consider just the company mailbox and not her personal one, and the inferred information would not be presented to other people than who the company wants to involve. But how to include privacy concerns is surely a challenge to be addressed in future work.

References

1. C. Hill, R. Yates, C. Jones, and S. L. Kogan, "Beyond predictable workflows: Enhancing productivity in artful business processes," *IBM Systems Journal*, vol. 45, no. 4, pp. 663–682, 2006.
2. P. Warren, N. Kings, I. Thurlow, J. Davies, T. Buerger, E. Simperl, C. Ruiz, J. M. Gomez-Perez, V. Ermolayev, R. Ghani, M. Tilly, T. Bösser, and A. Imtiaz, "Improving knowledge worker productivity - the active integrated approach," *BT Technology Journal*, vol. 26, no. 2, pp. 165–176, 2009.
3. W. M. P. van der Aalst, "The application of petri nets to workflow management," *Journal of Circuits, Systems, and Computers*, vol. 8, no. 1, pp. 21–66, 1998.
4. R. Agrawal, D. Gunopulos, and F. Leymann, "Mining process models from workflow logs," in *Advances in Database Technology – EDBT'98*.
5. J. E. Cook and A. L. Wolf, "Discovering models of software processes from event-based data," *ACM Trans. Softw. Eng. Methodol.*, vol. 7, no. 3, pp. 215–249, 1998.
6. W. M. P. van der Aalst, "Verification of workflow nets," in *ICATPN*, ser. Lecture Notes in Computer Science, P. Azéma and G. Balbo, Eds., vol. 1248. Springer, 1997, pp. 407–426.
7. W. M. P. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, 2004.

8. L. Wen, W. M. P. van der Aalst, J. Wang, and J. Sun, "Mining process models with non-free-choice constructs," *Data Min. Knowl. Discov.*, vol. 15, no. 2, pp. 145–180, 2007.
9. A. Weijters and W. van der Aalst, "Rediscovering workflow models from event-based data using little thumb," *Integrated Computer-Aided Engineering*, vol. 10, p. 2003, 2001.
10. A. K. Medeiros, A. J. Weijters, and W. M. Aalst, "Genetic process mining: an experimental evaluation," *Data Min. Knowl. Discov.*, vol. 14, no. 2, pp. 245–304, 2007.
11. W. van der Aalst, V. Rubin, H. Verbeek, B. van Dongen, E. Kindler, and C. Günther, "Process mining: a two-step approach to balance between underfitting and overfitting," *Software and Systems Modeling*, vol. 9, pp. 87–111, 2010.
12. J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev, "Synthesizing petri nets from state-based models," in *Computer-Aided Design, 1995. ICCAD-95. Digest of Technical Papers, 1995 IEEE/ACM International Conference on*, nov. 1995, pp. 164 –171.
13. ——, "Deriving petri nets from finite transition systems," *Computers, IEEE Transactions on*, vol. 47, no. 8, pp. 859 –882, aug. 1998.
14. J. Desel and W. Reisig, "The synthesis problem of petri nets," *Acta Informatica*, vol. 33, pp. 297–315, 1996.
15. X. Dong, A. Y. Halevy, and J. Madhavan, "Reference reconciliation in complex information spaces," in *SIGMOD Conference*, ACM, 2005, pp. 85–96.
16. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 1–16, 2007.
17. S. Chaudhuri, V. Ganti, and R. Motwani, "Robust identification of fuzzy duplicates," in *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, apr. 2005, pp. 865 – 876.
18. S. Guha, N. Koudas, A. Marathe, and D. Srivastava, "Merging the results of approximate match operations," in *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pp. 636–647.
19. N. Sarkas, A. Angel, N. Koudas, and D. Srivastava, "Efficient identification of coupled entities in document collections," in *ICDE 2010*, pp. 769–772.
20. D. Zardetto, M. Scannapieco, and T. Catarci, "Effective automated object matching," in *ICDE 2010*, pp. 757–768.
21. Z. Michalewicz, *Genetic algorithms + data structures = evolution programs (2nd, extended ed.)*. New York, NY, USA: Springer-Verlag New York, Inc., 1994.
22. A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining," *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, vol. 20, no. 1, pp. 19–62, May 2005.
23. M. W. Berry and M. Castellanos, *Survey of Text Mining II: Clustering, Classification, and Retrieval*, M. W. Berry and M. Castellanos, Eds. Springer, September 2007.
24. F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.
25. S. Sakurai and A. Suyama, "An e-mail analysis method based on text mining techniques," *Appl. Soft Comput.*, vol. 6, no. 1, pp. 62–71, 2005.
26. S. Sakurai, Y. Ichimura, A. Suyama, and R. Orihara, "Acquisition of a knowledge dictionary for a text mining system using an inductive learning method," in *Proceedings of IJCAI 2001 Workshop on Text Learning: Beyond Supervision*, 2001, pp. 45–52.

References

Di Ciccio, Claudio, Massimo Mecella, Monica Scannapieco, and Diego Zardetto (2011). "Groupware Mail Messages Analysis for Mining Collaborative Processes". In: *SEBD*. Ed. by Giansalvatore Mecca and Sergio Greco, pp. 397–404.

BibTeX

```
@InProceedings{   DiCiccio.etal/SEBD2011:GroupwareMailMessages,
  author        = {Di Ciccio, Claudio and Mecella, Massimo and Scannapieco,
                   Monica and Zardetto, Diego},
  title         = {Groupware Mail Messages Analysis for Mining Collaborative
                   Processes},
  booktitle     = {SEBD},
  year          = {2011},
  pages         = {397-404},
  crossref      = {SEBD2011}
}
@Proceedings{     SEBD2011,
  title         = {Sistemi Evoluti per Basi di Dati - {SEBD} 2011,
                   Proceedings of the Nineteenth Italian Symposium on Advanced
                   Database Systems, Maratea, Italy, June 26-29, 2011},
  year          = {2011},
  editor        = {Giansalvatore Mecca and Sergio Greco}
}
```