# Using Semantic Lifting for improving Process Mining: a Data Loss Prevention System case study

Antonia Azzini, Chiara Braghin, Ernesto Damiani, Francesco Zavatarelli

Dipartimento di Informatica
Università degli Studi di Milano, Italy
{antonia.azzini, chiara.braghin, ernesto.damiani,
francesco.zavatarelli}@unimi.it

**Abstract.** Process mining is a process management technique to extract knowledge from the event logs recorded by an information system. We show how applying an appropriate semantic lifting to the event and workflow log may help to discover the process that is actually being executed. In particular, we show how it is possible to extract not only knowledge about the structure of the process, but also to verify if some non-functional properties, such as security properties, hold during the process execution.

## 1 Introduction

Business Process Intelligence (BPI) is a research area that is quickly gaining interest and importance: it refers to the application of various measurement and analysis techniques both at design and at run-time in the area of business process management. In practice, BPI stands for an integrated set of tools for managing process execution quality by offering several features such as monitoring, analysis, discovery, control, optimization and prediction.

In particular, *process mining* is a process management technique to extract knowledge from the event logs recorded by an information system. It is often used for discovering processes if there is no a priori model, or for conformance analysis in case there is an a priori model that is compared with the event log in order to find out if there are discrepancies between the log and the model.

In this work, we focus our attention on process mining techniques based on the computation of frequencies among event dependencies to reconstruct the workflow of concurrent systems. In particular, we show how applying an appropriate *semantic lifting* to the event and workflow log may help to discover the process that is actually being executed. In the Web scenario, the term semantic lifting refers to the process of associating content items with suitable semantic objects as metadata to turn unstructured content items into semantic knowledge resources. In our case, the semantic lifting procedure corresponds to all the

transformations of low-level systems logs carried out in order to achieve a conceptual description of business process instances, without knowing the business process a priori.

To illustrate our proposal, we present a case study based on a *data loss prevention scenario* aiming to preventing the loss of critical information in companies. In order to describe our running example, we use a lightweight data representation model designed to support real time monitoring of business processes based on a shared vocabulary defined using open standard representations (RDF). We believe that the usage of RDF as modeling language allows independence and extremely flexible interoperability between applications.

The contributions of this paper are:

- an example on how semantic lifting may help to improve the discovering process during process mining;
- a definition of a Data Loss Prevention System in RDF, modeling a multi-level security policy based on the organizational boundaries (internal vs external actors and resources);
- an example on how, using semantic lifting in combination with standard process mining techniques during the discovery phase, it is possible to extract not only knowledge about the structure of the process, but also to verify if some non-functional properties, such as security properties, hold during the process execution.

This work is organized as follows. In Section 2, we introduce the semantic lifting approach and describe how it has been used so far; in Section 3, we give a short overview of the Resource Description Framework (RDF). Section 4 is the core of the paper, where we present the Data Loss scenario and we give some examples on how semantic lifting helps improving the investigation on the process. Section 5 concludes the paper.

## 2 Semantic Lifting: State of the Art

In the Web scenario, the term semantic lifting refers to the process of associating content items with suitable semantic objects as metadata to turn unstructured content items into semantic knowledge resources. As discussed in [3], by semantic lifting we refer to all the transformations of low-level systems logs carried out in order to achieve a conceptual description of business process instances. Typically, this procedure is implicitly done by converting data from the data storages of an information system to an event log format suitable for process monitoring [5]. We believe that this problem is orthogonal to the abstraction problem in process mining, dealing with different levels of abstraction when comparing events with modeled business activities [4]: our goal is to see how associating some semantics to an event from the log it is possible to extract better knowledge about some properties of the overall process, not to see which is the mapping between events and business activities/tasks.

So far, the term semantic lifting has been used in the context of model-driven software development. In [9], the authors proposed a technique for designing and implementing tool components which can semantically lift model differences arising among the tools. In particular, they used the term semantic lifting of differences to refer to the transformation of low-level changes to all the more conceptual descriptions of model modifications.

The literature [11] reports how the Business Process Management (BPM) usually operates at two main distinct levels, corresponding, respectively, to a management level, supporting business organizations in optimizing their operational processes, and a technology level, supporting IT users in process modeling and execution.

In these two levels, experts operate without a systematic interaction and cooperation, causing the well known problem of Business/IT alignment. In fact, one key problem is the alignment of different tools and methods used by the two communities (business and IT experts). In order to reduce the gap between these two levels, De Nicola and colleagues refer in [11] to *s*emantic technologies as an useful approach at supporting business process design, reengineering and maintenance of the business process, by highlighting some advantages related to the semantic lifting. The first one regards the support that the semantic lifting can give to business process design by a semantic alignment of a business process respect to a reference ontology. The semantic alignment can be achieved by performing consistency checking through the use of a reasoning engine. Then, the reengineering of a business process (BP) can be improved by providing suggestions to experts during the design phase of a BP, for example in finding alternative elements with semantic search and similarity reasoning over the business ontology. The authors also indicate, as another advantage, the possibility to support a BP maintenance by automatically checking the alignment between one of more business processes against the business ontology when the latter is modified. This can provide strong benefits since, for instance, a change in the company organization, could affect many business processes that need to be manually checked.

## 3 RDF to model Business Processes

Generally speaking, the Resource Description Framework (RDF) [7] corresponds to a standard vocabulary definition, which is at the basis of the Semantic Web vision, and it is composed by three elements: concepts, relations between concepts and attributes of concepts. These elements are modeled as a labelled oriented graph [6], defined by a set of triples $< s, p, o >$ where $s$ is subject, $p$ is predicate and $o$ is object, combined as shown in Figure 1.

New information is inserted into an RDF graph by adding new triples to the set. It is therefore easy to understand why such a representation can provide big benefits for real time business process analysis: data can be appended 'on the fly' to the existing one, and it will become part of the graph, available for any

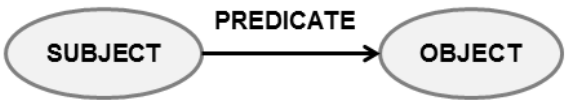


**Fig. 1.** RDF subject-object relation.

analytical application, without the need for reconfiguration or any other data preparation steps.

RDF standard vocabularies allow external applications to query data through SPARQL query language [12]. SPARQL is a standard query language for RDF graphs based on conjunctive queries on triple patterns, identifying paths in the RDF graph. Thus, queries can be seen as graph views. SPARQL is supported by most of the triples stores available.

Moreover, RDF provides a basic set of semantics that is used to define concepts, sub-concepts, relations, attributes, and can be extended easily with any domain-specific information. For this reason, it is an extremely generic data representation model that can be used in any domain.

In [10], the authors present a framework based on RDF for business process monitoring and analysis. They define an RDF model to represent a generic business process that can be easily extended in order to describe any specific business process by only extending the RDF vocabulary and adding new triples to the triple store. The model is used as a reference by both monitoring applications (i.e., applications producing the data to be analyzed) and analyzing tools. On one side, a process monitor creates and maintains the extension of the generic business process vocabulary either at start time, if the process is known a priori, or at runtime while capturing process execution data, if the process is not known. Process execution data is then saved as triples with respect to the extended model. On the other side, the analyzing tools may send SPARQL queries to the continuously updated process execution RDF graph.

Figure 2 shows the conceptual model of a generic business process, seen as a sequence of different tasks, each having a start/end time and possibly having zero or more sub-tasks. We will use this model to describe our running example.

## 4 Case study: a Data Loss Prevention System

Data loss is an error condition in information systems in which information is destroyed by failures or neglect in storage, transmission, or processing. Consider for example some different companies belonging to the same manufacturing supply chain and sharing business process critical data by using a file sharing server in order to access to the data. This scenario could expose the critical data to malicious users if access control is not implemented correctly. Indeed, an access control to such a server should be carried out by a security model, based on specific rules considering, for example, the user authentication for file sharing,

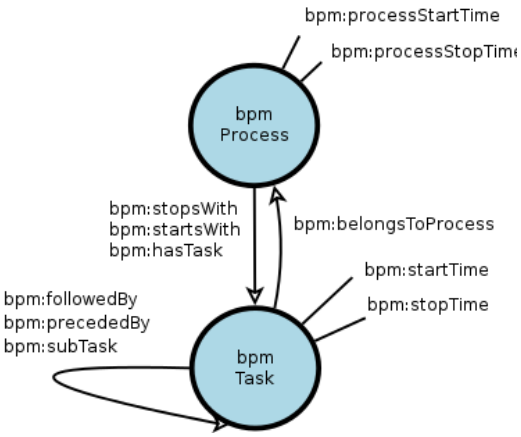


**Fig. 2.** RDF Representation of a generic business process.

security policies definition for users that have access rights to some confidential data, data checking before sending them to external companies that do not belong to the manufacturing supply chain considered, usage of authorized channels for data delivery, like company's e-mail, and so on. In order to prevent any kind of data loss, systems usually develop an intellectual ownership defense, also called *data-loss model*, tracking any action operated on a document. The process is able to highlight some security-related information, such as the economical value assigned to the outgoing intellectual ownership, or the number of 'confidential' data sent around, possibly to external destinations.

Protecting the confidentiality of information stored in a computer system, or transmitted over a public network is a relevant problem in computer security, called *information leakage*. The approach of information flow analysis involves performing a static analysis of the program with the aim of proving that there will not be leaks of sensitive information. The starting point in secure information flow analysis is the classification of program variables into different security levels (i.e., defining a multi-level security policy). In the simplest case, two levels are used: public (or low, L) and secret (or high, H). There is an information flow from object $x$ to object $y$ whenever the information stored in $x$ is transferred to, or used to derive information transferred to, object $y$. The main purpose is to prevent leak of sensitive information from an high variable to a lower one.

In our case study, we will consider the two security levels generated by the organizational boundaries (internal/external).

### 4.1 An RDF Model of the Data Loss case study

Our RDF model representation of the Data Loss case study is based on the lightweight RDF data model for business processes analysis carried out in [10] and briefly described in Section 3. As previously pointed out, the model does not contain any data, but it only provides a generic schema that the process

monitoring applications extend and instantiate. The Data Loss RDF model is then defined as an extension of the general schema and is depicted in Figure 3: it consists of the conceptual model of Figure 2 extended with domain specific concepts taken from the Data loss problem scenario. In particular, in our running example, we assume that the files shared between the companies in the same manufacturing chain are CAD files, thus the business process is named 'pCAD:CAD Process'.


**Fig. 3.** RDF Representation of the Data Loss Problem.

The main task is 'pCAD:UseFile', which creates a data file ('pCAD:File'), that can be used by a project manager ('pCAD:Employee'), which is an employee of one of the companies. All the concepts labeled 'dLoss' are the one which specify the security model to prevent Data Loss. The subclass called 'dLOSS:File' is assigned several attributes: the security level ('dLoss:ConfidentialFile'), and the economic value ('dLOSS:economicValue'), in order to be able to check information leakage or to compute the economical loss of the outgoing intellectual ownership. The destination of an operation on a file ('dLOSS:Destination') has three main attributes: 'dLOSS:InternalNetwork', 'dLOSS:ValueNetworkActor' and 'dLOSS:ExternalActor', the last specifying if the file has been shared with a user within the organizational boundaries or not.

As reported in Section 3, the usage of a framework based on an RDF triple store like [10] is specifically designed for integrating multiple source and supporting fast and continuous execution of SPARQL queries favouring the join between the execution processes and the monitoring.

### 4.2 Semantic Lifting for Mining a Data Loss Process

In this section, we show how an appropriate semantic lifting may help during the process mining phase. Process mining is the technique of distilling a struc-

tured process description from a set of real executions. To the sake of discussion, we limit our example to process mining algorithms that are based on detecting ordering relations among events to characterize a workflow execution log [14]. In particular, they build dependency/frequencies tables that are used to compare single executions in order to induce a reference model, or to verify the satisfiability of specific conditions on the order of executions of events. We assume that the reader is familiar with the following definitions that are common in this scenario [2].

**Workflow trace.** Let $E = \{e_1, e_2, \cdots, e_n\}$ be a set of events, then $t \in E^*$ is a *workflow (execution) trace.*

**Workflow log.** Let $E = \{e_1, e_2, \cdots, e_n\}$ be a set of events, then $W \subseteq E^*$ is a *workflow log.*

**Successor.** Let $W$ be a workflow log over $E$ and $a, b \in E$ be two events, then $b$ is a *successor* of $a$ (notation $a \prec_W b$) if and only if there is a trace $t \in W$ such that $t = \{e_1, e_2, \cdots, e_n\}$ with $e_i \equiv a$ and $e_{i+1} \equiv b$. Similarly, we use the notation $a \prec_W^n b$ to express that event $b$ is *successor* of event $a$ **by** $n$ **steps** (i.e., $e_i \equiv a$ and $e_{i+k} \equiv b$, with $1 < k \leq n$).

Notice that the successor relationship is rich enough to reveal many workflow properties since we can construct dependency/frequency tables that allow to verify the relations that constraint a set of log traces. However, in order to better characterize the significance of dependency between events, other measures, based on information theory, are adopted in the literature, such as for instance the J-Measure proposed by Smyth and Goodman [13], able to quantify the information content of a rule.

Table 1 shows a fragment of a workflow log possibly generated by a data loss prevention system tracking in-use actions based on the RDF model described in the previous section. The system reports all the events that generated a new status of a specific document. In particular, we assume that for each event it is specified: (*i*) the type of event (Create, Update, Share, Remove); (*ii*) the user performing the action on the file expressed by the email address; (*iii*) the timestamp spotlighting the end point (a system user, in our case) that achieved the control on the document at the end of the event which allow us to chronologically order the events; and (*iv*) the estimated value of the file (in the range: Low, Medium, High).

Following the approach in [14], we construct the dependency/frequency (D/F) table from the data log illustrated in Table 1. More in detail, the information contained in Table 2, are:

- the overall frequency of event $a$ (notation $\#a$);
- the frequency of event $a$ followed by event Create (C for short);
- the frequency of event $a$ followed by event Update (U for short) by 1, 2 and 3 steps;
- the frequency of event $a$ followed by event Share (S for short) by 1, 2 and 3 steps.

**Table 1.** An example of workflow log for the Data Loss case study.

| Event | User | Timestamp | Status | Estimated value |
|---|---|---|---|---|
| *File AAAA* | | | | |
| Create | userP@staff.org | 2012-11-09 T 11:20 | Draft | High |
| Update | userP@staff.org | 2012-11-09 T 19:20 | Draft | |
| Share | userA@staff.org | 2012-11-12 T 10:23 | Proposal | |
| Update | userA@staff.org | 2012-11-14 T 18:47 | Proposal | |
| Share | userP@staff.org | 2012-11-15 T 12:07 | Proposal | |
| Update | userP@staff.org | 2012-11-18 T 09:21 | Recommendation | |
| Share | userM@inc.org | 2012-11-18 T 14:31 | Recommendation | |
| *File AAAB* | | | | |
| Create | userF@staff.org | 2012-12-03 T 09:22 | Draft | Medium |
| Update | userF@staff.org | 2012-12-03 T 12:02 | Draft | |
| Update | userF@staff.org | 2012-12-03 T 17:34 | Draft | |
| Share | userV@staff.org | 2012-12-05 T 11:41 | Draft | |
| Share | userD@staff.org | 2012-12-05 T 11:41 | Proposal | |
| Update | userD@staff.org | 2012-12-08 T 10:36 | Proposal | |
| Update | userV@staff.org | 2012-12-08 T 16:29 | Proposal | |
| Share | userG@inc.org | 2012-12-10 T 08:09 | Proposal | |
| Update | userV@staff.org | 2012-12-10 T 18:38 | Recommendation | |
| *File AAAC* | | | | |
| Create | userV@staff.org | 2012-12-04 T 10:26 | Draft | Medium |
| Update | userV@staff.org | 2012-12-04 T 13:12 | Draft | |
| Update | userV@staff.org | 2012-12-05 T 10:12 | Draft | |
| Share | userA@staff.org | 2012-12-05 T 12:22 | Draft | |
| Share | userD@staff.org | 2012-12-06 T 14:51 | Proposal | |
| Share | userM@inc.org | 2012-12-07 T 10:31 | Proposal | |

Using this table we can observe that the following patterns hold in $W$: $Create \succ_W Update$ or $Create \succ_W^* Share$, that is, a file is always created before being updated or shared.

**Table 2.** An example of Dependency/Frequency based on the Successor relation.

| $a$ | $\#a$ | $a \prec$ C | $a \prec$ U | $a \prec^2$ U | $a \prec^3$ U | $a \prec$ S | $a \prec^2$ S | $a \prec^3$ S |
|---|---|---|---|---|---|---|---|---|
| Create | 3 | 0 | 3 | 2 | 1 | 0 | 1 | 2 |
| Update | 10 | 0 | 3 | 3 | 2 | 6 | 5 | 5 |
| Share | 9 | 0 | 4 | 2 | 2 | 3 | 3 | 1 |

Since a 'data-loss model' is typically aimed at detecting anomalous behaviors, the expected behavior in the form of unwanted behaviors (black-listing) or wanted behavior (white-listing) needs to be defined. This can be done by identifying behavioral patterns over the sequences of events that are normally registered in the workflow logs. We may, for instance, be interested in mining expected behavior for documents shared within and outside the boundaries of the organization. Still focusing our attention on the Share events which might cause unwanted information flows, we might be interested to see which are the users that most frequently share the documents with other users either inside or outside the boundaries. To this aim, a semantic lifting procedure can be applied to the log data for remodeling the representation of the process and allowing additional investigations.

A first semantic lifting can be done by applying the Data Loss model described in the previous section to our log, in order to distinguish among events where files are shared internally or externally to the organization. In our example, the lifting can be done by exploiting two data transformations rules expressed according to Equation 1. Data are then mapped to the model using standard techniques for mapping RDF data [8].

$$\begin{gathered} User \| [A-Z0-9\_.-] + @staff + . [A-Z] \{2,4\} \\ \rightarrow dLOSS : Internal \\ [A-Z0-9\_.-] + @inc + . [A-Z] \{2,4\} \\ \rightarrow dLOSS : External \end{gathered} \tag{1}$$

After applying the semantic lifting to the log, we are able to build and fill Table 3, where a Share event is rewritten as 'Share Internal' when the Share event is performed by an Internal user, otherwise the event is rewritten as a 'Share External' event. In this new dependences/frequencies among events, we observe that a new pattern holds: $ShareInternal \succ_W ShareInternal \succ_W ShareExternal$. Informally, we can interpret this pattern in the execution traces as the identification of an expected behavior about document sharing: before a document is shared externally to the organization it has to pass some (typically two) internal steps.

**Table 3.** D/F after a first semantic lifting: Sharing between Internal/External Users.

| $a$ | # | $a \prec$ SI | $a \prec^2$ SI | $a \prec$ SE | $a \prec^2$ SE |
|---|---|---|---|---|---|
| Share Internal (SI) | 6 | 3 | 0 | 3 | 3 |
| Share External (SE) | 3 | 0 | 0 | 0 | 0 |
| SI $\prec$ SI | 3 | 0 | 0 | 0 | 3 |

Another semantic lifting can be done by grouping together all the Share log events performed by the same user. Please notice that, as described in detail in Section 3, RDF allows to easily aggregate data by considering their shared properties. Moreover, SPARQL queries allow us to manipulate data to view them in the appropriate structural order, by defining, for example, events that are grouped and aggregated by different attributes. Table 4 reports the frequencies of these events, referring to an event `Share` with `userV@staff.org` as `SV`, `Share` with `userA@staff.org` as `SA`, and so on.

**Table 4.** D/F after another semantic lifting: Sharing events for specific Users.

| $a$ | # | $a \prec$ SA | $a \prec^2$ SA | $a \prec$ SD | $a \prec^2$ SD | $a \prec$ SG | $a \prec^2$ SG | $a \prec$ SM | $a \prec^2$ SM | $a \prec$ SP | $a \prec^2$ SP | $a \prec$ SV | $a \prec^2$ SV |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SA | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| SD | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| SG | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SM | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SP | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| SV | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

We can observe that the table is sparse, therefore few patterns can be proved to hold in $W$. In our example, for instance, we can derive that in only one case $SA \succ^2_W SM$, meaning that User `userA@staff.org` shares a document before the same document is shared by User `userM@staff.org`. We can also derive that User `userM@staff.org` is always the last to share the document, possibly meaning that he is at the bottom of the organization hierarchy or that he is an untrusted user (thing that is supported by the fact that it is an external user). Given the low frequency of both cases, the two conclusions we drew are not particularly relevant since they are not supported by a large number of traces. The sparsity of the table is typical of so called 'spaghetti-like processes', i.e., unstructured processes where recurrent event sequences are not so easily defined [1]. In this case, a semantic lifting procedure could be applied to the log data for remodeling the representation of the process and implementing additional investigations.

## 5 Conclusion

In this paper we showed how standard process mining techniques can be combined with semantic lifting procedures on the workflow logs in order to discover more precise workflow models from event-based data. Moreover, we highlighted the benefits using RDF as a modeling formalism by using it in our case study. This is just a first step to show the feasibility and the advantages of the approach. As a future work we plan to study how to automatize the process by exploiting the usage of RDF as a modeling language.

## Acknowledgment

This work was partly funded by the Italian Ministry of Economic Development under the Industria 2015 contract - `KITE.IT` project.

## References

1. Van der Aalst, W.M.P.: Process mining: Discovering and improving spaghetti and lasagna processes. Keynote Lecture, IEEE Symposium Series on Computational Intelligence (SSCI 2011)/IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011) (April 2011)
2. Van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow mining: a survey of issues and approaches. Data Knowl. Eng. 47(2), 237–267 (Nov 2003), http://dx.doi.org/10.1016/S0169-023X(03)00066-1
3. Azzini, A., Ceravolo, P.: Consistent process mining over big data triple stores. In: Proceedings of the IEEE International Conference on Big Data. p. to appear. IEEE Publisher, June 27-July 2, 2013, Santa Clara Marriott, CA, USA (2013)
4. Baier, T., Mendling, J.: Bridging abstraction layers in process mining by automated matching of events and activities. In: Daniel, F., Wang, J., Weber, B. (eds.) Business Process Management, Lecture Notes in Computer Science, vol. 8094, pp. 17–32. Springer Berlin Heidelberg (2013)
5. Buijs, J.: Mapping data sources to xes in a generic way, master's thesis (2010)
6. Carroll, J., Bizer, C., Hayes, P., Stickler, P.: Named graphs. Journal of Web Semantics 3(3) (2005)
7. Hayes, P., McBride, B.: Resource description framework (rdf) (2004), http://www.w3.org/
8. Hert, M., Reif, G., Gall, H.C.: A comparison of rdb-to-rdf mapping languages. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 25–32. I-Semantics '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2063518.2063522
9. Kehrer, T., Kelter, U., Taentzer, G.: A rule-based approach to the semantic lifting of model differences in the context of model versioning. In: Automated Software Engineering (ASE), 2011 26th IEEE/ACM International Conference on. pp. 163–172 (2011)
10. Leida, M., Majeed, B., Colombo, M., Chu, A.: Lightweight rdf data model for business processes analysis. Data-Driven Process Discovery and Analysis, Series: Lecture Notes in Business Information Processing 116 (2012)

11. Nicola, A.D., Mascio, T.D., Lezoche, M., Tagliano, F.: Semantic lifting of business process models. 2012 IEEE 16th International Enterprise Distributed Object Computing Conference Workshops 0, 120–126 (2008)
12. Prudhommeaux, E., Seaborne, A.: Sparql query language for rdf (2008), http://www.w3.org/
13. Smyth, P., Goodman, R.M.: Rule induction using information theory. Knowledge discovery in databases 1991 (1991)
14. Van Der Aalst, W., Van Hee, K.: Workflow management: models, methods, and systems. MIT press (2004)