Learning From the Past: Uncovering Design Process Models Using an Enriched Process Mining

Lijun Lan
Department of Mechanical Engineering,
National University of Singapore,
Singapore 117576

Ying Liu[1]
Mechanical and Manufacturing Engineering,
School of Engineering Cardiff University,
Cardiff CF24 3AA, UK
e-mail: LiuY81@Cardiff.ac.uk

Wen Feng Lu
Department of Mechanical Engineering,
National University of Singapore,
Singapore 117576

Design documents and design project footprints accumulated by corporate information technology systems have increasingly become valuable sources of evidence for design information and knowledge management. Identification and extraction of such embedded information and knowledge into a clear and usable format will greatly accelerate continuous learning from past design efforts for competitive product innovation and efficient design process management in future design projects. Most of the existing design information extraction systems focus on either organizing design documents for efficient retrieval or extracting relevant product information for product optimization. Different from traditional systems, this paper proposes a methodology of learning and extracting useful knowledge using past design project documents from design process perspective based on process mining techniques. Particularly different from conventional techniques that deal with timestamps or event logs only, a new process mining approach that is able to directly process textual data is proposed at the first stage of the proposed methodology. The outcome is a hierarchical process model that reveals the actual design process hidden behind a large amount of design documents and enables the connection of various design information from different perspectives. At the second stage, the discovered process model is analyzed to extract multifaceted knowledge patterns by applying a number of statistical analysis methods. The outcomes range from task dependency study from workflow analysis, identification of irregular task execution from performance analysis, cooperation pattern discovery from social net analysis to evaluation of personal contribution based on role analysis. Relying on the knowledge patterns extracted, lessons and best practices can be uncovered which offer great support to decision makers in managing any future design initiatives. The proposed methodology was tested using an email dataset from a university-hosted multiyear multidisciplinary design project.

[DOI: 10.1115/1.4039200]

Keywords: design informatics, machine learning, process mining, text mining, design information extraction, process analysis

1 Introduction

In the information age today, the advancement and widespread application of information management systems [1,2] that use textual databases to organize information have been archiving vast amounts of digital design documents at various stages of product design projects. Examples include customer requirements, computer-aided design (CAD) models, emails, chat logs, design forums, test reports, customer reviews, and repair reports. As these archival documents have objectively recorded the execution of past design projects, they have become the essential source of design information and empirical knowledge to assist decision makers in better managing future design projects and their corresponding processes. It is, therefore, considered that mining empirical information from historical design documents and reutilizing them in practical design work is one of the most important factors to enable modern enterprises to gain sustainable competitive edge [3].

Most of the existing design information extraction systems have been focusing on extracting product-relevant information from design documents such as CAD models and sketches for product optimization. However, design documents such as emails, conference minutes, and conversation transcripts, which contain invaluable process-relevant information, are mostly underutilized in the context of design information extraction. Different from product-relevant information which focuses on product structures and product functions, process-relevant design information is more about "How a product is designed," "who executes what tasks," and "who often work together." Based on such information patterns, successes or failures of past design projects can be learned and reutilized to support decision making at all stages of product development by means of suggesting promising problem solutions, evaluating possible alternatives, allocating the most suitable resources, and identifying bottlenecks for improvements. A detailed review of process-based design reuse was carried out by Baxter [4]. However, most of the reviewed methods heavily rely on human experience and judgement to construct a process model, which is often error-prone, time-consuming, or virtually impossible due to the length of the project and its breadth in terms of technical capacity and geographic coverage. To avoid or reduce the influence of human involvement, a promising opportunity relies on computational algorithms to automatically uncover critical process-relevant information, e.g., process models, from archived design project documents.

Process mining is a prevailing technique that looks inside the process by automatically extracting business workflow models from event logs recorded by business information systems [5].

[1] Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received March 29, 2017; final manuscript received November 23, 2017; published online February 27, 2018. Assoc. Editor: Harrison M. Kim.

Although process mining has been proved as an efficient tool that learns from the past for business process management, creating automatic approaches for mining design process models from archival design documents is still a major challenge for efficient process information reutilization in the context of product design. As most of the design data that record design process executions are semistructured or unstructured texts, traditional process mining approaches that depend on structured event logs become incompetent in the context of design process model discovery. Furthermore, as design processes are usually unpredictable and iterative [6], design task execution is rarely repeated exactly in the same form and manner. Therefore, traditional process mining approaches that attempt to model process behavior in a flat and linear model might produce very huge and complex models for design processes.

In our previous study, we have proposed a layered text mining system which aims to discover process model from design documents recording past design processes [7]. As an extension, this paper presents a methodology for learning critical process-relevant design knowledge from the past via an enriched process mining approach. It intends to become a supporting tool that could help decision makers to improve future projects based on the best practices learned from past design projects. In detail, the proposed methodology consists of two main stages: uncovering the design process model first and then enabling knowledge learning from the uncovered model. The first stage is designed to establish a hierarchical process model from the input design documents using an enriched process mining approach. Different from the existing process mining techniques which deal with event logs and timestamps, the inputs in our study are archived design documents written in a natural language (in this case, English). In addition, by focusing on modeling the documented design process in a hierarchical and modular manner, the proposed approach is able to reduce the model complexity that is caused by the flexible nature of design process. The second stage focuses more on analyzing the uncovered process model so as to enable design information and empirical knowledge learning from multiple perspectives, e.g., the actual execution trace/route via workflow analysis, bottleneck via performance analysis, cooperation via social network analysis, and individual contributions via role analysis. Such studies would help designers making practical and efficient decisions in future design projects. For proposal validation, the effectiveness of the proposed methodology is demonstrated with the help of a case study taken from a real university-hosted design project.

The remainder of this paper is structured as follows: Section 2 reviews related works; Section 3 describes the first stage of the proposed methodology and different methods incorporated within the proposed process mining approach; Section 4 presents a number of statistical analysis methods used in the second stage. Section 5 reports the real-life case study; Section 6 discusses possible extensions and future work; and Sec. 7 concludes.

2 Related Work

2.1 Design Information Extraction. Design information extraction has its root in linguistics and data mining, with a particular focus on extracting high-quality design information in the form of patterns and terms from design documents by means of natural language processing and data mining. Typical design information includes customer requirements, design rationales, technology trends, product structures, problem solutions, and resource allocation. Through design information extraction, information contained within design documents can be made more accessible for assisting decision-makers in making more doable and efficient decisions, which aim to optimize product design or speed up product development processes.

Due to the easy access of CAD documents, a significant body of the early research work on design information extraction has focused on retrieving and reusing past CAD models by extracting parametric associations. Some of them use color and texture as main features to retrieve similar CAD drawings from an image database [8,9], while others treat CAD models as structured text files and use text mining techniques to represent CAD models as vectors of identifiers [10,11]. Based on the retrieved CAD models, design information relating to product geometry can be reused to speed up product development process via reusing the associated manufacturing processes [12], thus reducing the time required to generate a process plan. However, these CAD-based information extraction and reutilization systems are only suitable for solving geometric problems, which relate more to product structure.

Besides the product geometry embedded in CAD models, there is wealth of nongeometric design information embedded in other types of design documents such as patents [13], customer reviews [14], repair verbatim [15], production configuration data [3], and communication transcripts [16]. Two significant research streams are discovery of technology trends from patents and extraction of customer opinions from online reviews [13,17]. Both the technology trends and the customer opinions could inspire designers to search and identify solutions for product optimization. For example, in the ISAL (issue, solution, and artifact) system, a series of text mining algorithms were specifically designed to automatically discover design rationales from patent documents for an engineering design purpose. With a similar purpose of market-driven technology innovation, potential product concepts of solar-lighting devices were identified from a collection of domain-specific patents [18]. Another example is automatically translating customer reviews into engineering characteristics for quality function deployment [14].

The literature review on design information extraction indicates that most of the existing works have put attention on information relating to product, but extracting and reusing the information relating to design process has little work. Design process models, as an integrated part of design information, can be reused for decision making at various stages of design processes. Imperative efforts are needed to explore the potential of automatically extracting process relevant information like process models from historical design documents.

2.2 Process Mining. Process mining, also known as event mining or workflow mining, is a general methodology used to diagnose business processes by discovering models (e.g., Petri net, business process model and notation, and event graph models) that describe reality from historical event data [19]. The business process models discovered can be compared with a priori models to check whether reality, as recorded in the event log, conforms to the business specifications [20], or be used for simulation and performance analysis [21]. Traditionally, process mining has been focusing on control-flow discovery; that is, automatically discovering the causal dependencies or execution patterns between activities from enactment logs [22–24]. In recent years, as techniques have matured, process mining has been applied successfully in a wide range of real cases, e.g., shipbuilding industry [25], risk management [26], financial service [27], and healthcare processes [28].

Although traditional process mining techniques are able to discover high quality models from logs of well-structured processes, they usually return "spaghetti-like" models when applied to logs of unstructured processes [29]. The most essential reason is that traditional process mining approaches aim to unify all the behaviors recorded in event logs in a unique and flat model. This strategy is not suitable for unstructured processes, where process executions greatly differ from each other. As a remedy, Gunther and van der Aalst [29] proposed a fuzzy mining to simplify the discovered model with the concept of roadmap abstraction. Maggi et al. [30,31] used a semistructured process scheme referred to as "declarative workflow" to present unstructured processes with a set of constraints that state the rules among activities. Diamantini et al. [32] employed hierarchical graph clustering to

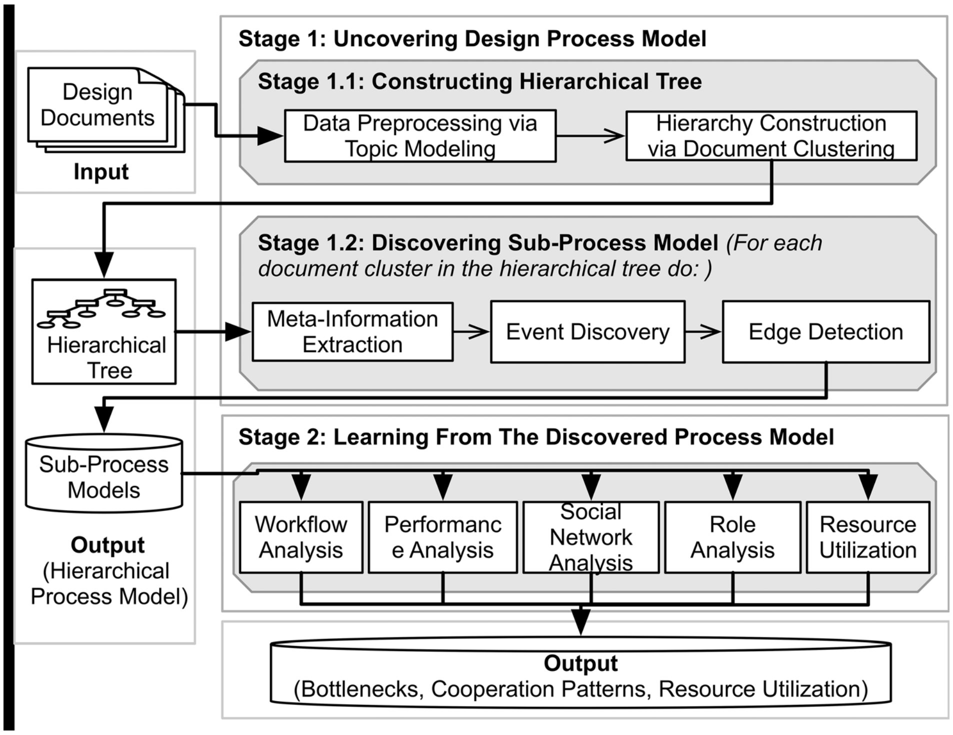

Stage 1: Uncovering Design Process Model
Stage 1.1: Constructing Hierarchical Tree
Design Documents
Input
Data Preprocessing via Topic Modeling
Hierarchy Construction via Document Clustering
Stage 1.2: Discovering Sub-Process Model (For each document cluster in the hierarchical tree do:)
Hierarchical Tree
Meta-Information Extraction
Event Discovery
Edge Detection
Sub-Process Models
Output (Hierarchical Process Model)
Stage 2: Learning From The Discovered Process Model
Workflow Analysis
Performance Analysis
Social Network Analysis
Role Analysis
Resource Utilization
Output (Bottlenecks, Cooperation Patterns, Resource Utilization)

Fig. 1 The methodology of learning from archival design project documents

identify subprocesses, which reflect meaningful collaboration work practices.

As traditional process mining approaches lack the ability of handling unstructured data, mining process model from natural language texts has been attaining more and more attention in recent years. Sinha and Paradkar [33] utilize use cases as source documents and presented a text analysis approach for semi-automatically transforming use cases into business process models. Friedrich et al. [34] combines the existing tools from natural language processing and augmented them with an anaphora resolution mechanism to generate business process model and notation models from process descriptions. Most of these process mining approaches follow a similar mining scheme, which consists of three steps: syntactic analysis which focuses on tokenization and par-of-speech tagging, semantic analysis which detects actions and actors using semantic dictionaries and knowledge bases, and model generation which discovers sequence flows through predefined signal words. One major limitation of these approaches is that the input text has to describe a model sequentially, and the statements in the descriptions must relate to process model. Furthermore, creating such descriptions requires extra manual efforts.

To summarize, because design process is often flexible, and the execution of a design process is usually recorded in a text-rich format, it has become imperative to research suitable techniques, such as process mining, for the discovery of underlying design process models. Furthermore, to address the difficulties currently faced, such techniques should not only be able to handle the textual data as the evidence left alongside a design process, but also be able to model its process structure which is inherently flexible.

3 Uncovering Design Process Model From Design Project Documents

Figure 1 depicts the proposed methodology of learning process-relevant information and knowledge from past design documents based on an enriched process mining approach. As shown in Fig. 1, the starting point of the whole system is a set of design documents, which record the process executions of a past design project in natural language format. Based on the design documents, the embedded design process is uncovered and analyzed in two stages: uncovering the design process model and learning from the discovered process model.

The goal of the first stage is to uncover a hierarchical process model, which consists of a hierarchical tree and a set of subprocess models, using the proposed process mining approach. The obtained hierarchical tree decomposes the embedded design process into several functional modules. The subprocess models present the detailed execution traces of the modules in the hierarchical tree.

The second stage aims to distill multifaceted knowledge patterns from the discovered process model via statistical analysis, such as workflow analysis for uncovering task dependencies, performance analysis for detecting potential bottlenecks or irregular task executions, social network analysis for discovering cooperation patterns, as well as resource utilization and role analysis for estimating the degree of individual contributions.

Details about the first stage of uncovering design process model are reported in this section, whereas details on the second stage are presented in Sec. 4.

3.1 Constructing Hierarchical Tree. A top-down clustering approach based on document content is specifically designed to surface a modular representation of the embedded design process. The heuristic is that design documents with similar content are likely to be relevant to the same design task in reality. Therefore, the proposed approach decomposes the input design documents into clusters based on their content similarity. The content similarity of any two documents relies on the overlapping of their topic distributions. Furthermore, in order to get a hierarchical representation of the underlying design process, the decomposition procedure proceeds in a top-down manner, until desired homogeneousness is achieved. As a result, the hierarchical representation is a tree, and each node of the tree corresponds to a functional module, within which more detailed, homogeneous executions could be observed from the corresponding document cluster.

Figure 2 describes the top-down clustering approach for constructing the hierarchical tree in detail. The meanings of some fundamental notions are defined as follows:

```
Algorithm 1 Top-down clustering for constructing hierarchical tree
Inputs: $\boldsymbol{D}$ is the document collection, $\gamma$ is the maximum homogeneousness and $\mathscr{P}$ indicates the maximum depth.
1:  Procedure TOPDOWN_CLUSTERING($D$, $\gamma$, $\mathscr{P}$)
2:    For each $D_i$ in $D$ do:
3:      Topic modeling: $D_i = (h_i, \ldots, h_H)$
4     Initialization: $C := \{D\}$, $M := \{(D, \varnothing)\}$, $T := \{\varnothing\}$
5:    While $|C| > 0$ do:
6:      $c^p := pop(C)$ //Select and delete a document cluster from C
7:      If $homogeneity(c^p) < \gamma$ and $depth(c^p) < \mathscr{P}$ do:
8:        $C^{new} = HCA\_clustering(c^p, D_{c^p})$ //Cluster a module
9:        For each $c^s$ in $C^{new}$ do:
10:         $m^{new} := (c^s, \varnothing)$, $M := m^{new} \cup M$
11:         $T := (m^{new}, m^{c^p}) \cup T$
12:       $C := C^{new} \cup C$
```

Fig. 2 Top-down clustering for constructing hierarchical tree

- C is a set of document clusters;
- M is a set of functional modules, each module (sw_i, C_i) corresponds to a subprocess model sw_i mined from a document cluster C_i; and
- $T \subset \{M \times M\}$ is a tree that organizes M in a hierarchical structure.

The algorithm starts by representing the documents using a set of latent topics. As deep belief networks (DBN) [35,36] have been witnessed to perform well in both learning and fast inferring per-document topic distributions, the DBN-based topic modeling approach [16] is employed to obtain the top distribution for each input document. A typical DBN model is a deep neural network consisting of one input layer of observation, one output layer of reconstruction, and several hidden layers. It transforms a document D_i from a word-frequency vector into a topic-probability vector $D_i = (h_1, \ldots, h_H)$, where H is the number of topics detected from the entire document archive D, and $h_j \in [0, 1]$ is the probability that the jth topic appears in a document.

Based on the per-document-topic representation, the algorithm starts constructing the process hierarchy by initializing it with a root node $M := \{(D, \varnothing)\}$, which is the whole document set. Next, in per iteration shown in lines 5–12, one document cluster with the least homogeneous content is selected from C and partitioned into smaller ones using the top-down hierarchical clustering algorithm. Equation (1) computes the content homogeneity of a document cluster C_i, which is inversely proportional to the average distance from the cluster members to the cluster center

$$\text{homogeneity}(C_i) = 1 - \sum_{d \in C_i} \text{dist}\left(D_d, \overline{D_{ci}}\right) / |C_i| \tag{1}$$

where D_d is the per-document-topic distribution, $\overline{D_{ci}}$ is the average per-document-topic distribution of C_i, $\text{dist}\left(D_d, \overline{D_{ci}}\right)$ measures the cosine distance between any two topic distributions, and $|C_i|$ is the cluster size.

After each decomposition, a set of new hierarchical relations is created in T (shown in line 11). The whole decomposition process can be iterated until all the leave modules in T are homogeneous enough, producing a hierarchical structure of the process in T.

3.2 Discovering Subprocess Model. The second stage aims to mine subprocess models for the modules in the generated hierarchical tree from the corresponding document clusters. Let *sw* be a subprocess model. The formal representation of *sw* is a tuple (O, A, E), where O is a finite set of fundamental elements that point to physical objects, such as people, organizations, tools, and locations, A is a finite set of design events which consists of several physical objects in O, and $E \subseteq A \times A$ defines the potential sequence in which design events have been executed. Based on this definition, the procedure of subprocess model discovery is further decomposed into three steps: meta-information extraction for O, event discovery for A, and edge detection for E.

3.2.1 Meta-Information Extraction. The focus of meta-information extraction is extracting special writing expressions, called as named entities (NEs), from the inputted design texts. The extracted NEs might point to some physical objects that had been involved in the target design process. In this work, seven types of process relevant NEs are considered, namely, tasks/activities (TE), timestamps (SE), persons (PE), organizations (OE), locations (LE), input/output information (IE), and techniques/tools (ME). A hybrid named entity recognition (NER) approach is proposed to recognize the above NEs from texts. By treating all the noun phrases (NP) as candidate NEs, the proposed NER approach first generates a small set of seed NEs from the candidate NPs via rule matching. Next, through learning more complex features from the seed NEs, the proposed NER approach expands more general NEs from the remaining candidate NEs. The motivation for using the integration of rule matching and machine learning is to keep human intervention at the minimum.

3.2.1.1 Seed entity generation. The rules for matching seed entities are created based on the concept of "speech acts." Originally, "speech acts" are defined as "illocutionary" verbal utterances that have a performative function to present a speaker's intentions, such as promising, ordering, requesting, and inviting [23]. Based on this theory, this paper considers that noun phrases associated with special verbs or nouns might point to process relevant objects with high confidence. Examples of verbal utterances include "submit," "complete," "use," and "software." This paper calls such verbs and nouns as speech act words. Each entity type owns a speech act dictionary W_E and a set of matching rules formed on W_E. The preliminary set of speech act words in W_E is manually selected from a set of randomly selected sentences. To get a more general W_E, the preliminary W_E is then expanded by including more synonyms using WordNet,[2] which is a large lexical database of English.

[2] Princeton University "About WordNet." WordNet. Princeton University, 2010. http://wordnet.princeton.edu

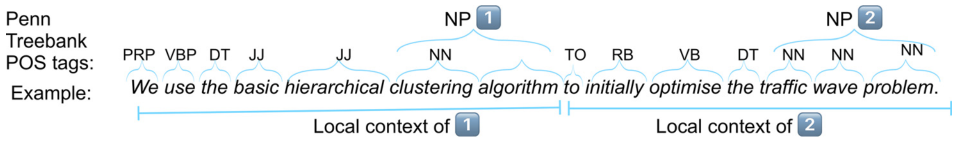


Fig. 3 Example of local context

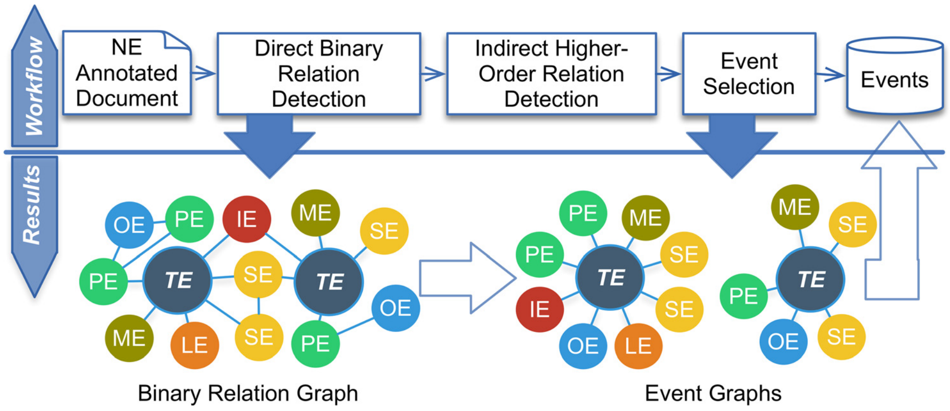


Fig. 4 Workflow of design event discovery

Give the speech act dictionary W_E, seed NE is defined as noun phrases that are associated with speech act verbs or contain speech act nouns in W_E. Here, the open library, Stanford CoreNLP [37] which provides a set of natural language analysis tools, is used to find the candidate noun phrases in the texts.

3.2.1.2 Entity expansion. Based on the seed entities detected, a kernerlized support vector machine (SVM) classifier is trained to retrieve more general NEs from the remaining candidate NEs. As kernel function can save time and effort of explicitly selecting features into the feature space, a string kernel based on surrounding words [16] is employed to measure the similarity between two NEs.

It is the normal case that one sentence might mention several NEs simultaneously. For distinguishing NEs appearing in the same sentence, only the surrounding words located in a local context of a seed or candidate NE are used to compute the string kernel. As highlighted in Fig. 3, the local context of an NE is defined as the words from the end of its preceding NP to its own last word. Furthermore, due to the common sense that words which have a shorter distance to a NE play a more significant role in conveying the meaning of the NE, all the words in the local context are weighted according to their instance to the head word of a NE

$$\mathrm{con}(\mathrm{ne}) = (w_1 c_1, \ldots, w_V c_V) \tag{2}$$

$$w_i = 1 - d_i/\mathrm{len}(\mathrm{NE}) \tag{3}$$

where c_i indicates the ith word in the local context of ne, w_i is the weights, d is the number of words between the ith word and the head word of ne, and len(ne) is number of words in the local context of ne.

Based on Eqs. (2) and (3), the kernel function for training the SVM classifier is defined as

$$\mathrm{kernel}(\mathrm{ne}_1, \mathrm{ne}_2) = \sum_{i=1}^{i=\mathrm{len}(\mathrm{ne}_1)} \sum_{j=1}^{j=\mathrm{len}(\mathrm{ne}_2)} \min(w_i, w_j)\,\mathrm{sign}(c_i, c_j)/(\mathrm{num}_c + 1) \tag{4}$$

where $\mathrm{sign}(c_i, c_j)$ returns 1 or 0, indicating whether c_i and c_j are two identical words, and num_c is the number of words shared in the local contexts of two NEs.

Using the above kernel function, a SVM classifier is trained on the seed entities and applied to predict the entity type of the remaining candidate NEs.

3.2.2 Event Discovery. The second step aims to detect design events from design documents by identifying the semantic relations among the recognized NEs. Let $\mathrm{NE}_{\mathrm{TYPE}} = \{\mathrm{TE}, \mathrm{PE}, \mathrm{SE}, \mathrm{OE}, \mathrm{LE}, \mathrm{IE}, \mathrm{ME}\}$ be the entity types in the first step, a design event is defined as a graph $\mathrm{EG} = (V, v_0, t_s, t_e E)$, where

- V—each vertex $v \in V$ denotes a NE, and the entity type of v belongs to $\mathrm{NE}_{\mathrm{TYPE}}$;
- v_0—the graph is centered at v_0, $v_0 \in V$, and the entity type of v_0 must be TE (task entity);
- t_s, t_e–$t_s, t_e \in V$ are the starting and ending time of an event, and their entity type must be SE (time entity);
- E—each edge $e \in E$ denotes a relation between a normal vertex and the center vertex v_0, therefore, $E \in \{v_0 \times V\}$.

According to the above definition, a design event can be viewed as a higher-order relation that is centered at a task entity, and all the other related NEs connect to the central task entity directly or indirectly. Two problems here are the number of NEs in design events is varying, and event graphs might overlap on some vertices because design events could share some resources. To tackle the two problems, a graph partition based method of higher-order relation extraction is proposed. Figure 4 shows the workflow of the proposed approach. The basic idea is factorizing the higher-order relation in a design event graph into several binary relations, and reconstructing the design event by finding the maximal cliques around each task entity based on the binary relations.

3.2.2.1 Direct binary relation detection. It is the normal case that two entities mentioned in the same sentence are more likely to be related. Based on this assumption, the step of binary relation detection is to find pairs of NEs that are mentioned in the same sentence and have a semantic relation. A rule-based pattern matching approach is used to match binary relations sentence by sentence. The rules include

- Rule 1: two entities must be mentioned in the same clause.
- Rule 2: two entities are directly connected in the sentence dependency tree.

- Rule 3: the type of two entities must be consistent with one of the binary relation types predefined via expert knowledge.
- Rule 4: the sentence or clause is in present tense.
- Rule 5: no negative words (e.g., don't, not) exists between two entities.

It is worth to mention that rule 3 is designed to eliminate unpractical binary relations that provide no or less significant information for event detection, for example, relations between two location entities and two time entities. In addition, rule 4 is introduced to find design events that are being done or will be done, and rule 5 is for eliminating negative relations.

This step would produce a binary relation graph $G = (V, E)$. The vertices in V are the NEs mentioned in a document, the edges in E are the binary relations between the mentioned NEs, and the weight of each edge, $w(e|e \in E)$, is the frequency of the corresponding binary relation.

3.2.2.2 Indirect higher-order relation detection. The step of indirect higher-order relation detection aims to partition the binary relation graph $G = (V, E)$ by finding the maximal clique around each entity in G. Each maximal clique is a candidate event, and maximal cliques around different task entities can overlap on some vertices.

First, a clique that is centered at a task entity v_0 is defined as a subgraph of G in the form of $G' = (V', v_0, E')$, where $V' \subseteq V$, $E' \subseteq E$, $v_0 \subseteq V'$, $\text{type}(v_0) = \text{TE}$, and for each $v \in V' - v_0$, there is at least one path from v to v_0. Therefore, a maximal clique that is centered at v_0 is defined as $\text{MG} = (V^{\text{MG}}, v_0, E^{\text{MG}})$ and there is no other clique $G' = (V', v_0, E')$ that $\text{density}(G') > \text{density}(\text{MG})$. The function $\text{density}(G')$ shown in Eq. (5) computes the density of a clique using the mean of the edge weights

$$\text{density}(G') = \left(\sum_{e \in E'} w(e)\right) / |V'| \tag{5}$$

Next, the maximal clique around each task entity is greedily detected by expanding an initial clique in all the directions that the clique density increases. The initial clique is a subgraph in which all the entities have a direct relation to the central task entity. New nodes are added to the initial clique if they are connected to the initial clique and the density of the new clique is larger than the density of the old one. The process of clique expansion stops when no more nodes could be added. By this means, nodes with weak connections to their neighbors would be eliminated, and the number of the entities in an event is determined by the local density of the binary relation graph.

3.2.2.3 Event selection. The candidate events obtained in the last step might contain very small cliques that might be noises. To filter the noisy cliques off, the step of event selection first ranks the candidate events detected from a single document by weighting them according to the sum over their edge weights. Next, a cutoff ρ is then used to select candidate events whose weights fall into the ρ% top rank. In this way, only maximal cliques that have a larger number of nodes or stronger edges remain as design events.

For simplifying the discovered design event graph, all the paths from the central task node to other indirectly connected nodes are replaced by single edges. The weight of the new edges is the smallest edge weight in the corresponding paths.

Finally, the starting and beginning time nodes of each design event are simply set as the minimal and maximal time indicated by the time entity nodes. If no time entity node is included in a clique, the creation time of the corresponding document is used in place.

3.2.3 Edge Detection. Let the discovered design events be the task nodes in the subprocess model, the step of edge detection is to identify the precedence relations among the task nodes. Given two design events (e.g., a_i and a_j), the possibility that a precedence relation exists is estimated by how close the two events were executed

$$\text{pr}(a_i \rightarrow_g a_j) = 1 - d/g \tag{6}$$

where g is the size of the time window and d is the time interval, which is determined by the starting time of a_i and the ending time of a_j. Based on $\text{pr}(a_i \rightarrow_g a_j)$, the relation between a_i and a_j is classified as follows:

- $a_i \neq\neq a_j$: there is no relation if $\text{pr}(a_i \rightarrow_g a_j) \in (\propto, 0] \cup (1, \propto)$;
- $a_i == a_j$: e_i is parallel with e_j if $\text{pr}(a_i \rightarrow_g a_j)$ equals 1, which means two events are executed at the same time;
- $a_i \rightarrow a_j$: e_j is executed following e_i if $\text{pr}(a_i \rightarrow_g a_j) \in (0, 1)$.

Only when the precedence relation $a_i \rightarrow a_j$ is detected, a corresponding edge is added into the subprocess model. It is noteworthy that by using the time criteria, not only the direct relations $a_i \rightarrow a_j$ and $a_j \rightarrow a_r$ but also the long-distant relation $a_i \rightarrow a_r$ are taken into account as long as a_i and a_r are executed close enough. In this way, two events that are highly related but disturbed by a third event can be reconnected.

4 Learning From the Discovered Process Model

The second stage of the proposed methodology proceeds to distill multifaceted design information of empirical knowledge patterns from the hierarchical process model using statistical analysis methods. In detail, the discovered process model is viewed in three correlated dimensions, consisting of design tasks/events, personnel, and time. Different analysis methods are applied to individual dimensions or the combination of any two dimensions to mine particular process information that could be helpful in solving special problems. Figure 5 illustrates this analysis process in a three-dimensional model. Referring to Fig. 5, the personnel dimension is the people involved in the discovered process, and the task dimension refers to subprocess models in the hierarchical tree or the small design events that constitute the subprocess models. As shown in Fig. 6, the discovered process model is analyzed from five perspectives: workflow analysis, performance analysis, role analysis, social network analysis, and resource utilization analysis.

4.1 Workflow Analysis. The workflow aspect of the uncovered process model provides information regarding the question: "what does the actual process look like?"; by analyzing the relationship between the subprocess models or the design events. This is achieved through two aspects. First, the hierarchical tree captures the subordination relationships between the functional modules, which reflects how the entire design process is iteratively decomposed to several smaller design tasks. Second, the subprocess model within each module reflects the detailed execution traces that a specific design problem is solved. Such a hierarchical

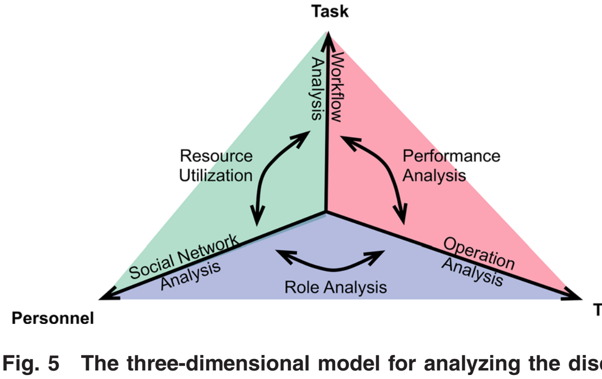


Fig. 5 The three-dimensional model for analyzing the discovered process model

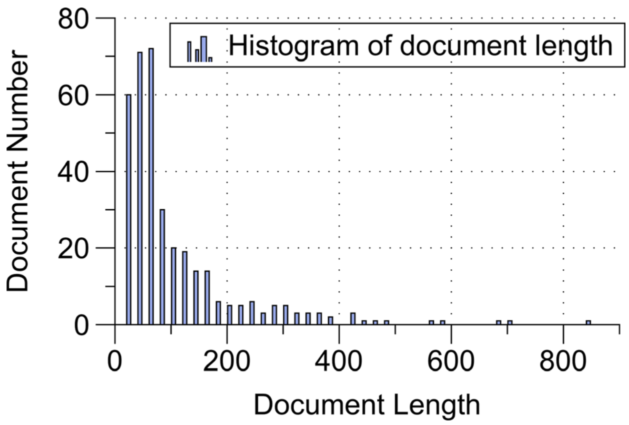


Fig. 6 Histogram of document length

representation is able to help decision-makers to quickly locate and understand the parts they are interested in.

4.2 Performance Analysis. The performance aspect analyzes the subprocess models according to their execution time. This is helpful to answer questions like: "are there any irregular task executions or bottlenecks in the actual process?" In product design process, irregular executions or bottlenecks usually are design tasks that slow down the whole design process. Identifying irregular executions or bottlenecks allows decision-makers to determine the area where problem occurs and identifies the root causes, so as to avoid the same mistakes in a new design project. To identify irregular executions, subprocess models are compared in a dotted chart, which represents the subprocess models or their subordinative events in the vertical axis, and the corresponding execution time in the horizontal axis. Based on the dotted chart, irregular executions and bottlenecks might be subprocess models that have an extremely long duration or were suspended frequently. The execution traces related to the irregular subprocess models should be carefully inspected, so as to detect the actual root causes such as lacking resources, waiting for the outputs of other design tasks, and having an operator who needs training.

4.3 Role Analysis. The role analysis aims to determine the relative value of the people involved by measuring and comparing their contributions to the whole design process. By focusing on the interaction between personnel and time, generalists who were always active throughout the entire design process could be recognized as core participants, whereas people who only participated in some specific design events could be recognized as specialists in the field relevant to those design events. The dotted chart is again used to perform role analysis. The vertical axis represents the people included in the process model, the horizontal axis indicates the time, and each dot denotes that a people is involved in a design event at a time. The density of the dots reflects the contribution of the people to the whole design process. Based on the doted chart, the key personnel from a similar past project can be quickly identified and considered for a new design project.

4.4 Social Network Analysis. The social network aspect analyzes the relationship between the involved people, aiming to provide information about "who is typically working together?" The social network graph is used to measure and visualize the connections between the project participants based on their joint events. Based on the connections, the project participants can be clustered into groups. From an organizational perspective, these discovered groups reflect different departments cooperating on the design process. Furthermore, the degree of incoming edges within a group measures the possibility that a people is the leader of this group. Similarly, the degree of incoming edges across groups measures the possibility that a people is a manager of the design process as managers usually interact with people from different departments. Equation (7) calculates the possibility that a participant is a manager

$$s\_\mathrm{manager}(p) = \frac{\mathrm{fr\_num}(\mathrm{nei}(p) - \mathrm{coop}(p))}{\mathrm{fr\_num}(P)} * \sum_{p' \in \mathrm{nei}(p) - \mathrm{coop}(p)} w(p,p') \tag{7}$$

where P denotes all the involved people, $\mathrm{nei}(p)$ indicates the people who are directly connected to p in the social network graph, $\mathrm{coop}(p)$ are the people in the same group with p, $w(p,p')$ is the interaction strength of two people, and function fr_num() calculates the number of unique group labels in a set of participants.

4.5 Human Resource Utilization. People involved in past design projects and their levels of knowledge and skills are usually valuable resources that can be transformed to produce benefit in future design projects. To extract information about human resources, such as "who executed what tasks?" and "to what degree was a people involved in a task?," the human resource utilization aspect analyzes the relationship between the personnel and the subprocess models. A histogram is created for each subprocess model to compare the engagement of the involved people. This could provide decision-makers helpful guidance when allocating the most suitable people to a similar new design task, so as to improve the human resource utilization in the new project.

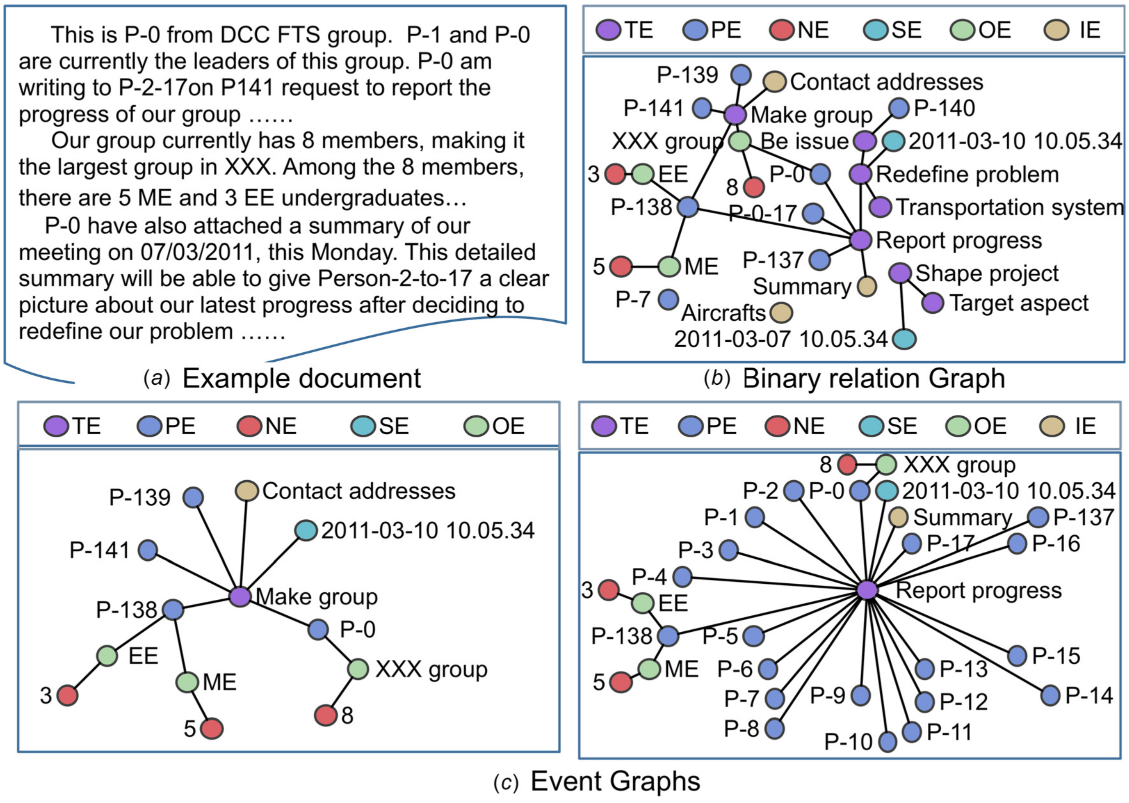
5 Case Study

All the algorithms integrated in the proposed process mining approach were implemented in PYTHON. The entire process mining approach was demonstrated and verified on a real case study of a university-hosted design project named "traffic wave project." It focused on designing a traffic control system to ease the traffic congestion on expressway and published the study results in a conference paper [38]. This project had eight core participants, including students and professors from two different disciplines. Throughout the design process, they frequently contacted a lot of people from engineering design, vehicle design, control, model building, and external companies. In addition, as this project is only one of the correlated subprojects of a bigger project, core participants from other subprojects were also involved. The whole design process lasted about 2 years, from March 2011 to February 2013.

5.1 Dataset and Evaluation. The example data were a set of emails collected from the traffic wave project. Throughout the design process, all of the participants always sent a copy to a specific common account when they used emails to exchange and discuss their opinions. This culminated in a total of 569 emails that were collected and saved in a MS Outlook file. Each email contains information about the design tasks discussed in the email body, the involved people are mentioned as either the email sender/receiver or in the email body, and the time is indicated by the creation time of the emails.

The original dataset was cleaned by removing the duplicates in the reply thread, resulting in 357 remaining emails. The personal pronouns in the email body were replaced by the names of the people in the TO, FROM, and CC fields of the corresponding emails. To give a deeper impression of the cleaned email dataset, Fig. 6 plots the histogram of email length.

To validate how well the discovered process model conform to reality, two senior participants and one junior participant from the traffic wave project were requested to sketch the originating process model embedded in the email dataset prior. Next, the quality of the discovered process model was assessed by mapping it back to the originating process model with the experts' help. To assess the correctness of the discovered knowledge patterns, every



This is P-0 from DCC FTS group. P-1 and P-0 are currently the leaders of this group. P-0 am writing to P-2-17on P141 request to report the progress of our group ……

Our group currently has 8 members, making it the largest group in XXX. Among the 8 members, there are 5 ME and 3 EE undergraduates…

P-0 have also attached a summary of our meeting on 07/03/2011, this Monday. This detailed summary will be able to give Person-2-to-17 a clear picture about our latest progress after deciding to redefine our problem ……

(*a*) Example document

(*b*) Binary relation Graph

(*c*) Event Graphs

Fig. 7 Examples of design event

knowledge pattern was checked with the three interviewed participants. A knowledge pattern was right if it was in accordance to the experts' experience.

5.2 Experimental Results

5.2.1 Overview of the Hierarchical Process Model. Three hundred sentences were randomly selected to manually seed the speech act words, and 13,734 NEs including 191 unique personal names were recognized by the hybrid NER approach. By setting the cutoff parameter ρ as 0.8, 661 design events were detected by the event detection approach, and 41 subprocess models were constructed in the hierarchical tree.

To give an intuitive feeling of the discovered design event, Fig. 7 illustrates an example of event detection. Figure 7(*a*) gives an example document which consists of multiple sentences. Figure 7(*b*) shows the binary relation graph constructed from it. In Fig. 7(*b*), seven entities were recognized as task entities, i.e., "make group," "report progress," "redefine problem," "be issue," "adopt transportation system," "target aspect," and "shape project." Figure 7(*c*) illustrates two event graphs detected from the binary relation graph in Fig. 7(*b*). From the entities contained in the event graphs, it can also be observed that entities within a design event are usually mentioned in different sentences, rather than all in a single sentence.

5.2.2 Workflow Analysis. Figure 8 compares the automated subprocesses with the originating tasks that were given by the interviewed participants. All the subprocesses listed in Fig. 8 are named according to its most frequent event. Subprocesses for which the interviewed participants can find a relevant originating task are highlighted and connected to their originating tasks in the middle column. Figure 8 shows that 39 of the 41 subprocesses have a counterpart in the originating tasks. In contrast, eight of the nine originating tasks are connected to at least one subprocess. These findings indicate that the automated subprocesses have an actual reflection of the originating tasks, but in a more detailed view.

Figure 9 illustrates a segment of the discovered hierarchical process model. However, due to the space limitation, Fig. 9 only shows three subprocesses. The rectangular nodes present small design events, and the arrows indicate their workflows within a subprocess. From the names of the design events, it can be observed that the three subprocesses were all concerning the execution of a presentation, but with regard to different aspects. The first subprocess in Fig. 9 shows a very clear workflow of scheduling and rescheduling the presentation date, the second subprocess is about making the presentation, and the third subprocess reflects the procedure of making assessment after the presentation. According to the feedback from the interviewed participants, the three subprocess models correctly reflect the procedure that the undergraduate participants of this project did the presentation for their final year project using the achievements of this traffic wave project.

5.2.3 Performance Analysis. Figure 10(*a*) visualizes the dotted chart of the subprocess models. The horizontal axis indicates the time, and the vertical axis presents the 41 subprocesses listed in Fig. 8. The dots are the events belonging to each subprocess. The dotted chart in Fig. 10(*a*) shows that most subprocesses proceeded concurrently, e.g., tasks 1–3 and tasks 4–7. From Fig. 10(*a*), it can also be observed that the design activities in most subprocesses have been more or less temporally suspended. Subprocesses that have been suspended frequently or for a long time might be executed irregularly. Based on this assumption, Fig. 10(*a*) shows that there are six subprocesses that might have irregular executions, i.e., task 3, task 6, task 9, task 12, task 16, and task 18.

Figure 10(*b*) plots the temporal event throughput, which calculates the number of events executed in a short period. Continuous periods that have temporal event throughputs larger than a threshold of average throughput are identified as busy periods, otherwise inactive periods. According to Fig. 10(*b*), this traffic wave project

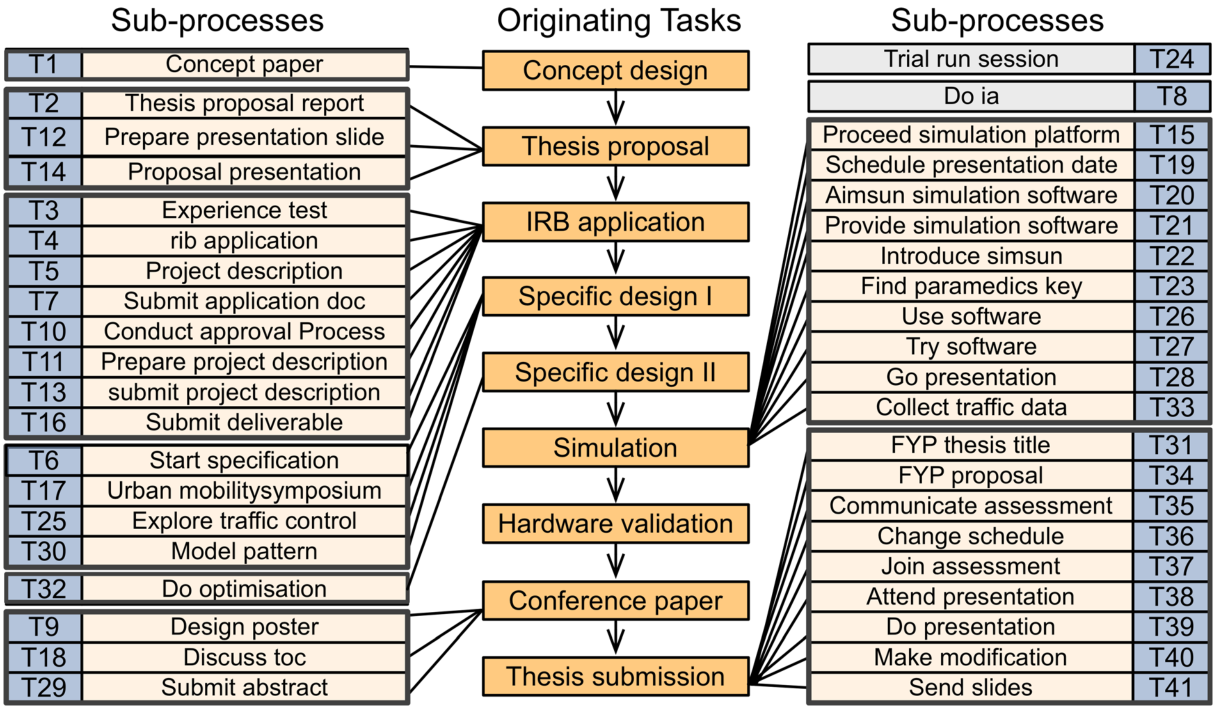


Fig. 8 Comparison between the automated subprocesses and the originating tasks

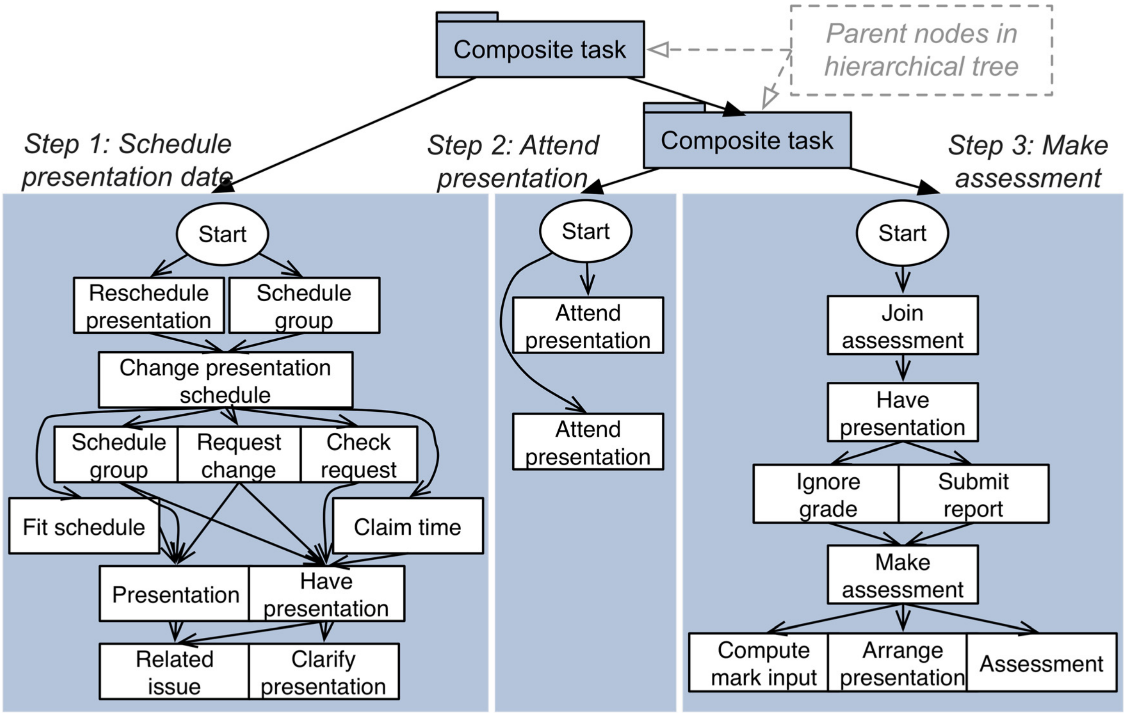


Fig. 9 A segment of the hierarchical process model

had four relatively busy periods and three relatively inactive periods. The design tasks started during the inactive periods might be causes that impede the project progress. In Fig. 10, the subprocesses started during the first inactive period are tasks 4–9, the subprocesses started during the second inactive period are tasks 15–17, and no subprocesses are started during the third inactive period. The interviewed participants explained that during the first inactive period, the whole design project was temporally suspended to apply an approval from a related organization, and they were not familiar about the application procedure, therefore slowing down the design progress. For the second inactive period, it was explained that the core participants spent a long time in contacting the manufacturers of different simulation tools before they found a suitable one, and task 15 is related to this procedure.

Taking the above analyses together, the detected irregular subprocesses or bottlenecks indicate the areas where problems that might slow down project progress occur. Identifying irregular executions or bottlenecks and their potential root causes allow decision-makers to be aware of such issues and avoid them in a new design project if it bears similar nature.

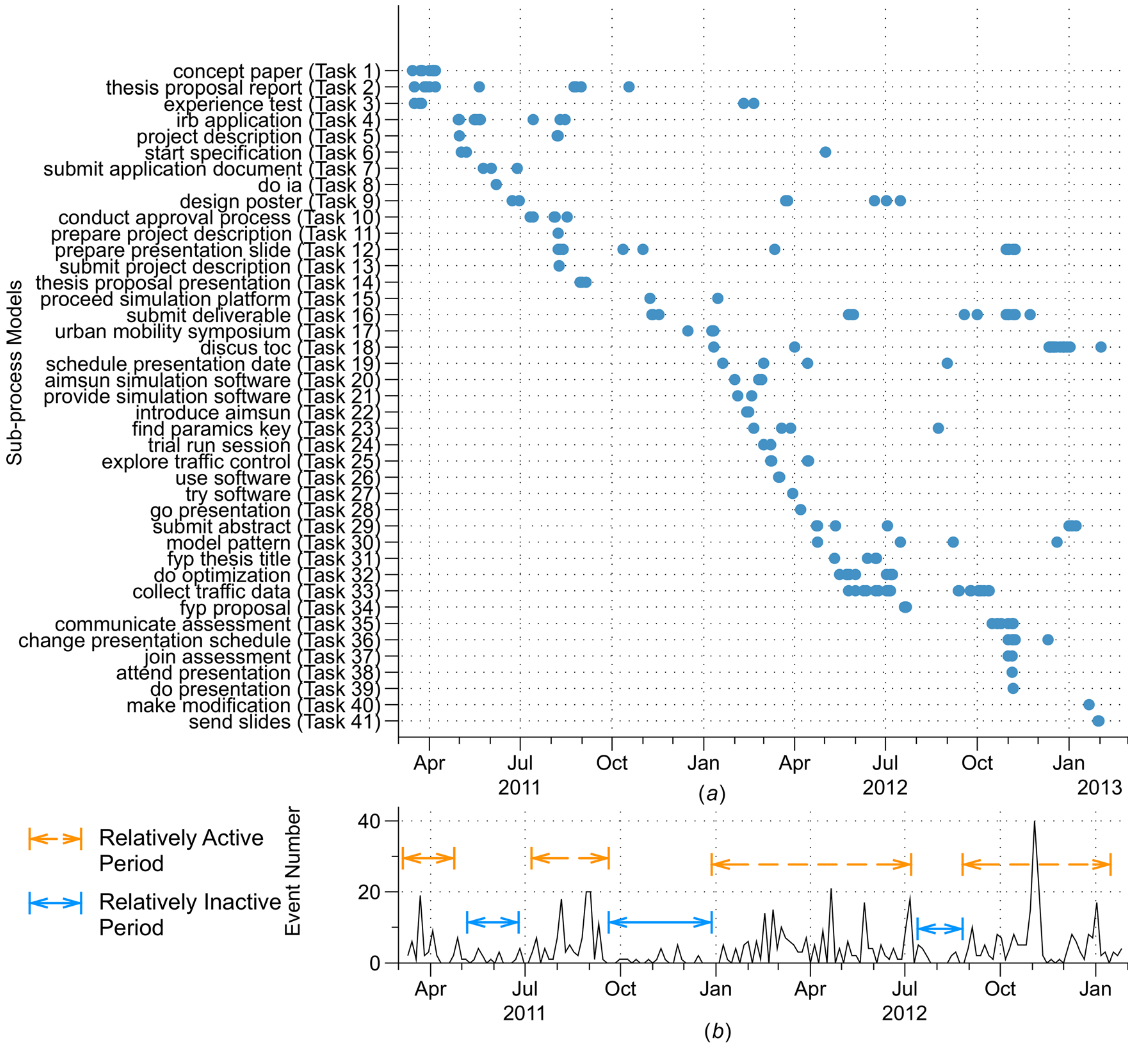


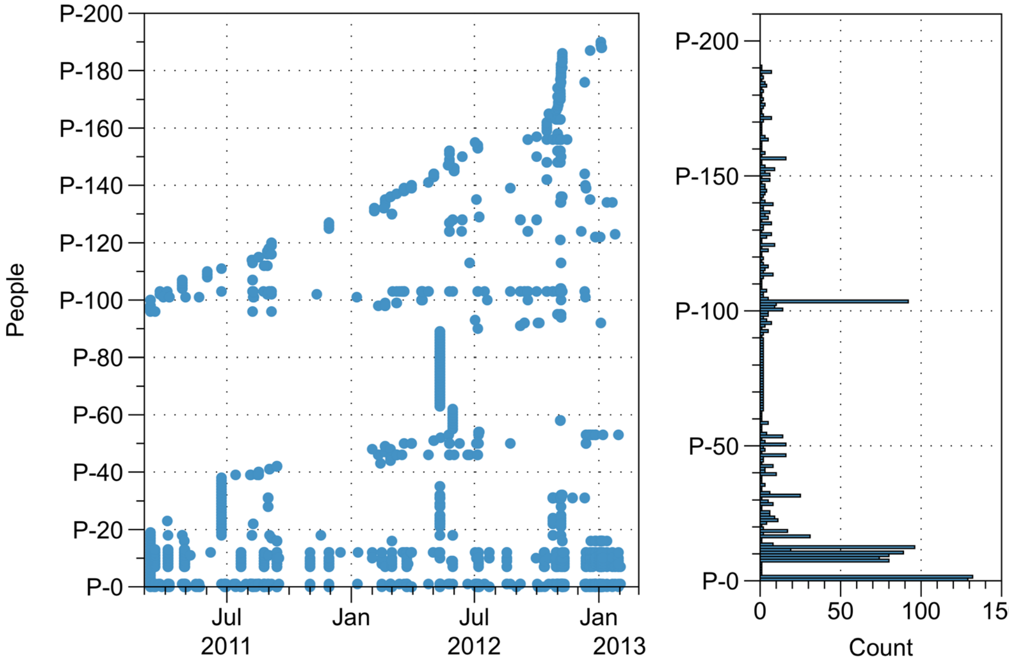
Fig. 10 Performance analysis: (*a*) dotted chart of subprocesses and (*b*) temporal event number

5.2.4 Role Analysis. Figure 11(*a*) plots the dotted chart for comparing the relative engagement of the people involved in the traffic wave project. The vertical axis represents the 191 people detected in the discovered process model. The dots indicate that people are involved in some events at some time. The local dot density in Fig. 11(*a*) reflects the temporal engagement of the corresponding people in a continuous period. For example, Fig. 11(*a*) shows that P60-80 joined the project very late, and P160-180 were active at the beginning but withdrew midway. Such participants who were involved only in a short period might be specialist in handling some special design tasks. The global dot density of each line indicates the overall engagement of the people in the entire design process. To visualize the global dot density, Fig. 11(*b*) plots the number of the dots in each line. Figure 11(*b*) shows that P0-1 and P7-12 might be core participants of the traffic wave project as they had the largest global dot densities and were continuously active throughout the entire design process.

5.2.5 Social Network Analysis. Figure 12 shows four social networks formed under different conditions. Figure 12(*a*) visualizes the interaction behaviors of all the 191 participants. As Fig. 11 shows that a significant portion of the participants only participated in a small number of events, Figs. 12(*b*) and 12(*c*) filter out participants less actively engaged so as to help decision-makers to focus more on the behaviors of the actively involved members. In the four social networks, the nodes in the same color mean people who are clustered in the same cliques, except for the red nodes which denote people who fail to join any clique. The solid edges depict the interaction within a clique, while the dashed edges reflect the interaction across different cliques. The interaction strength of any two connected nodes is measured by the number of their joint events.

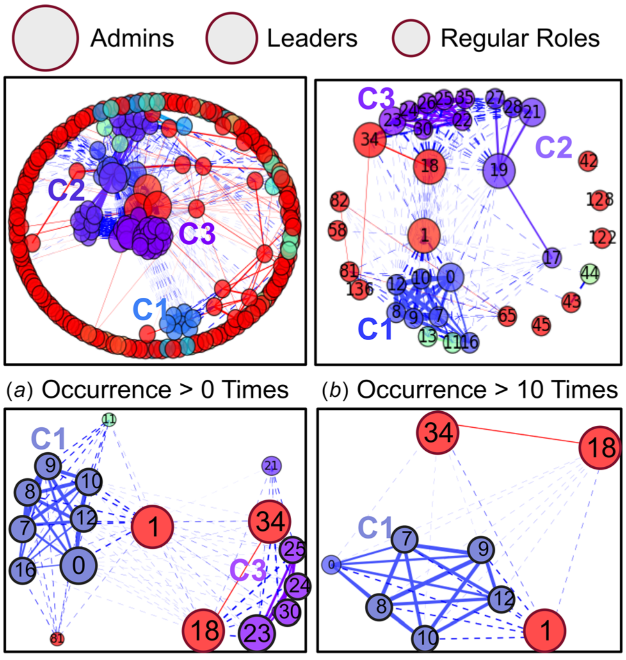
The graphs in Fig. 12 reveal the existence of three big cliques, within which people interacted frequently. Among the three cliques, C1, including P0, P7-10, P12, and P16, corresponds to the core participants of this project. This is consistent with the observation obtained from the role analysis in Fig. 11. The nodes filtered out by the graphs in Figs. 12(*b*)–12(*d*) indicate that people in C2 and C3 engaged less actively than the people in C1. This finding is consistent with the feedback that people in C2 and C3 were not the main participants of this project, but participants from other sister subprojects.

Inspecting the degree of incoming edges across different cliques reveals that there are four participants who frequently interact with people from different cliques, which indicates that they occupy a kind of administrative position. They are P1, P18, P19, and P34, and are denoted by the four biggest nodes in Fig. 12. In addition, Fig. 12 also highlights originators who have the highest degree of incoming edges within a clique. They are P0, P21, and P23, who occupy a kind of leadership position. This observation was confirmed by the interviewed participants; that the people denoted as admins are four professors who supervised different subprojects, and P0 is the student leader of the traffic wave project.



(*a*) Dotted chart (*b*) Overall engagement

Fig. 11 Role analysis via dotted chart



(*a*) Occurrence > 0 Times (*b*) Occurrence > 10 Times (*c*) Occurrence > 30 Times (*d*) Occurrence > 70 Times

Fig. 12 Social network analysis

5.2.6 Human Resource Utilization. Figure 13 illustrates two histograms that compare the percentage engagement of the people involved in two example subprocesses. The engagement degree of the participants is measured by the number of events they executed in a subprocess.

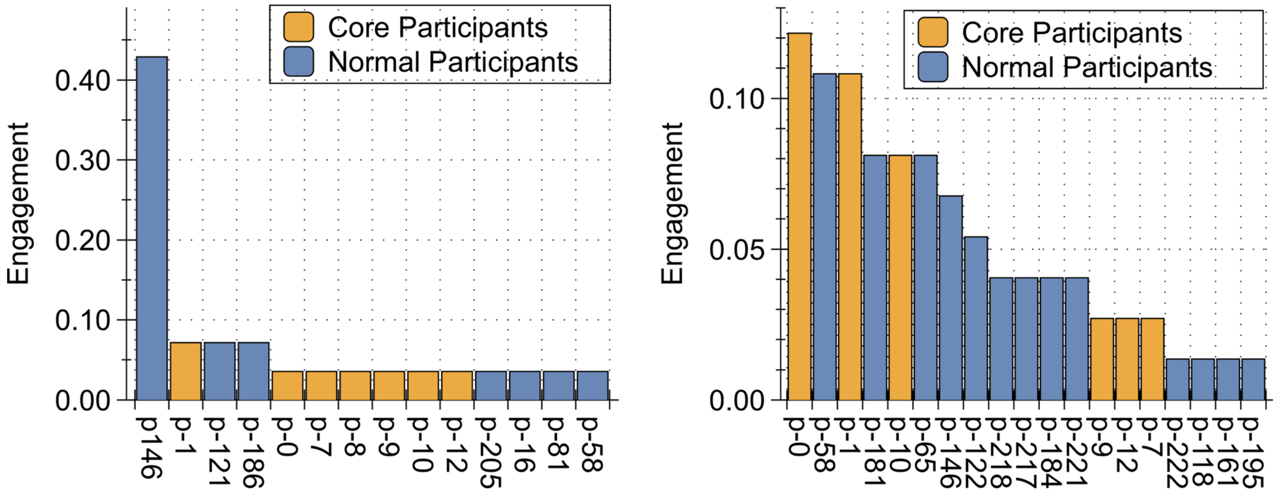
Referring to Fig. 13, 14 people are involved in the first subprocess, and 19 people in the second subprocess. Figure 13 also shows that all the core members included in the clique C1 of Fig. 12, i.e., P0-1, P7-10, P12, and P16, are involved in both subprocesses more or less. Another interesting observation can be made in Fig. 13(*a*) is that a noncore participant, P146, executed mostly a half of the events of the first subprocess, while the most active core member only executed 7% of the events. In contrast, the histogram in Fig. 13(*b*) reveals that the core members played a major role in the second subprocess as two of the top-three active participants are from the core members, i.e., P0 and P1.

6 Discussion

The validations with the experts' help have proven the efficacy of the proposed process mining approach. With respect to the workflow analysis, the discussion with the experts revealed that the discovered process model indeed represented the innate character of their processes. Moreover, the hierarchical structure allows people to focus only on the most relevant part of the process. This is especially helpful when the complete process is flexible and not so straightforward for junior personnel who are new to the project.

For the performance analysis, it was confirmed that the major task reflected by subprocesses, namely, T4, T5, T7, T10, and T11, indeed slowed down the whole project for several months. It was also stated that the inputs of the next major task reflected by T12 an T14 in Fig. 10 had no dependence on the outputs of T4, T5, T7, T10, and T11. In this case, the entire project time could be reduced by at least 3 months if T12 and T14 were started simultaneously with T4.

In the discussion about the role analysis results, the interviewed participants were somehow surprised that 191 people were involved throughout the project as there were only eight core members at the beginning. When given the social network graphs, the three participants could name different cliques and recognize the clique consisting of the core members. It is also interesting to be pointed out that three of the four participants recognized as admins, i.e., P1, P18, and P34, failed to join any cliques in Fig. 10. What this means for social network analysis is that one cannot focus only on people who interact with each other most frequently. Instead, some participants like admins might have less interactions with regular participants, but play a significant role in making all the participants work together. Knowing different kinds of cooperation patterns would help decision-makers to



(*a*) Participants in Sub-process of "Modeling Patterns"

(*b*) Participants in Sub-process of "Collecting Traffic Data"

Fig. 13 Examples of human resource utilization

allocate the most suitable human resources in the future design projects.

Additionally, the proposed process mining approach can help decision-makers gain a deeper understanding of a past design project from a more objective point of view. Because the process models are directly discovered from real-life design documents, they respect the reality and reduce the human bias introduced by conventional approaches such as surveys, interviews, and discussions. Furthermore, the discovered process model can be further analyzed to detect good practices or bad experiences such as irregular executions, delays, and bottlenecks in actual executions.

The discussion with the core participants also revealed several possibilities for future improvement and extension. The current work used project emails as the data source for case study validation. There might be situation that some design tasks were less-frequently discussed via email correspondence, but recorded by other types of design documents, such as conference minutes, progress reports, and conversation transcripts. For example, Fig. 8 shows that the process model discovered from the email dataset provides rare information about the originating task of hardware validation. Concerning this problem, the proposed approach itself is universal to other types of design documents. Therefore, as long as more design documents can be provided, the proposed approach could lead to a more comprehensive process model and other major performance patterns concerned.

The subprocesses shown in Fig. 9 also reflect some iterative design events. For example, the student participants scheduled the presentation date first. Next, they rescheduled the presentation date for several times. Such loops and iterations play a key role in measuring the efficiency of a design process and could help decision-makers to find solutions to facilitate design processes. Therefore, creating automatic approaches for identifying loops and iterations in the design process presents the opportunities to improve the ability of the discovered process model at supporting decision making.

At a more macrolevel, the proposed process mining approach also introduces the possibility of managing and retrieving past design documents in a structured, graphic manner. For example, by representing a design project as the process model uncovered from its archival documents, past design projects can be compared according to the structure similarity of their process models. This is a critical step to search for and reutilize interesting information from large quantities of past design projects.

7 Conclusions

This paper presented a methodology for learning from the historical design documents based on process mining. The proposed methodology was developed in two main stages: the discovery of a design process model from archival design project documents and learning multifaceted knowledge patterns from the discovered process models. Novelties of the proposed methodology include (1) proposing a new process mining approach with the capability of handling textual data; (2) capturing the flexibility of a design process via a hierarchical and modular representation; and (3) applying statistical analysis methods to learn valuable knowledge patterns from the uncovered process model. The proposed methodology has been tested using an email dataset collected from a university-level design project. The results provided evidence that the proposed approach can not only correctly reveal the actual executions of past design processes, but also return meaningful knowledge patterns to support future design project and process management.

References

[1] Kotinurmi, P., Laesvuori, H., Jokinen, K., and Soininen, T., 2004, "Integrating Design Document Management Systems Using the Rosettanet E-Business Framework," 6th International Conference on Enterprise Information Systems (ICEIS 2004), Porto, Portugal, Apr. 14–17, pp. 502–509.

[2] Linas, G., and Romualdas, B., 2006, "Electronic Document Management in Building Design," J. Civ. Eng. Manage., **12**(2), pp. 103–108.

[3] Efthymiou, K., Sipsas, K., Mourtzis, D., and Chryssolouris, G., 2015, "On Knowledge Reuse for Manufacturing Systems Design and Planning: A Semantic Technology Approach," CIRP J. Manuf. Sci. Technol., **8**, pp. 1–11.

[4] Baxter, D., Gao, J., Case, K., Harding, J., Young, B., Cochrane, S., and Dani, S., 2007, "An Engineering Design Knowledge Reuse Methodology Using Process Modelling," Res. Eng. Des., **18**(1), pp. 37–48.

[5] van der Aalst, W. M. P., Weijters, T., and Maruster, L., 2004, "Workflow Mining: Discovering Process Models From Event Logs," IEEE Trans. Knowl. Data Eng., **16**(9), pp. 1128–1142.

[6] Browning, T. R., Fricke, E., and Negele, H., 2006, "Key Concepts in Modeling Product Development Processes," Syst. Eng., **9**(2), pp. 104–128.

[7] Lan, L., Liu, Y., and Lu, W. F., 2016, "Discovering a Hierarchical Design Process Model Using Text Mining," ASME Paper No. DETC2016-59829.

[8] Tao, S., Huang, Z., Ma, L., Guo, S., Wang, S., and Xie, Y., 2013, "Partial Retrieval of CAD Models Based on Local Surface Region Decomposition," Comput.-Aided Des., **45**(11), pp. 1239–1252.

[9] Tao, S., Wang, S., and Chen, A., 2017, "3D CAD Solid Model Retrieval Based on Region Segmentation," Multimedia Tools Appl., **76**(1), pp. 103–121.

[10] Sivakumar, S., and Dhanalakshmi, V., 2013, "An Approach Towards the Integration of CAD/CAM/CAI Through STEP File Using Feature Extraction for Cylindrical Parts," Int. J. Comput. Integr. Manuf., **26**(6), pp. 561–570.

[11] Yu, W. D., and Hsu, J. Y., 2013, "Content-Based Text Mining Technique for Retrieval of CAD Documents," Autom. Constr., **31**, pp. 65–74.
[12] Huang, R., Zhang, S., Bai, X., Xu, C., and Huang, B., 2015, "An Effective Subpart Retrieval Approach of 3D CAD Models for Manufacturing Process Reuse," Comput. Ind., **67**, pp. 38–53.
[13] Liang, Y., Liu, Y., Kwong, C., and Lee, W., 2012, "Learning the 'Whys': Discovering Design Rationale Using Text Mining—An Algorithm Perspective," Comput.-Aided Des., **44**(10), pp. 916–930.
[14] Jin, J., Ji, P., and Liu, Y., 2015, "Translating Online Customer Opinions Into Engineering Characteristics in QFD: A Probabilistic Language Analysis Approach," Eng. Appl. Artif. Intell., **41**, pp. 115–127.
[15] Rajpathak, D. G., 2013, "An Ontology Based Text Mining System for Knowledge Discovery From the Diagnosis Data in the Automotive Domain," Comput. Ind., **64**(5), pp. 565–580.
[16] Lan, L., Liu, Y., Lu, W., and Alghamdi, A., 2015, "Automatic Discovery of Design Task Structure Using Deep Belief Nets," ASME Paper No. DETC2015-47369.
[17] Liu, Y., Liang, Y., Kwong, C. K., and Lee, W. B., 2010, "A New Design Rationale Representation Model for Rationale Mining," ASME J. Comput. Inf. Sci. Eng., **10**(3), p. 031009.
[18] Jin, G., Jeong, Y., and Yoon, B., 2015, "Technology-Driven Roadmaps for Identifying New Product/Market Opportunities: Use of Text Mining and Quality Function Deployment," Adv. Eng. Inf., **29**(1), pp. 126–138.
[19] Jans, M., van der Werf, J. M., Lybaert, N., and Vanhoof, K., 2011, "A Business Process Mining Application for Internal Transaction Fraud Mitigation," Expert Syst. Appl., **38**(10), pp. 13351–13359.
[20] da Cruz, J. I. B., and Ruiz, D. D., 2011, "Conformance Analysis on Software Development: An Experience With Process Mining," Int. J. Bus. Process Integr. Manage., **5**(2), pp. 109–120.
[21] Luengo, D., and Sepúlveda, M., 2011, "Applying Clustering in Process Mining to Find Different Versions of a Business Process That Changes Over Time," International Conference on Business Process Management (BPM), Clermont-Ferrand, France, Aug. 29–Sept. 2, pp. 153–158.
[22] Agrawal, R., Gunopulos, D., and Leymann, F., 1998, "Mining Process Models From Workflow Logs," Sixth International Conference on Extending Database Technology: Advances in Database Technology (ETBT), Valencia, Spain, Mar. 23–27, pp. 469–483.
[23] Tiwari, A., Turner, C. J., and Majeed, B., 2008, "A Review of Business Process Mining: State-of-the-Art and Future Trends," Bus. Process Manage. J., **14**(1), pp. 5–22.
[24] Li, J., OuYang, J., and Feng, M., 2012, "A Heuristic Genetic Process Mining Algorithm," Seventh International Conference on Computational Intelligence and Security (CIS), Hainan, China, Dec. 3–4, pp. 15–19.
[25] Seung-kyung, L., Bongseok, K., Minhoe, H., Sungzoon, C., Sungkyu, P., and Daehyung, L., 2013, "Mining Transportation Logs for Understanding the After-Assembly Block Manufacturing Process in the Shipbuilding Industry," Expert Syst. Appl., **40**(1), pp. 83–95.
[26] Caron, F., Vanthienen, J., and Baesens, B., 2013, "A Comprehensive Investigation of the Applicability of Process Mining Techniques for Enterprise Risk Management," Comput. Ind., **64**(4), pp. 464–475.
[27] De Weerdt, J., Schupp, A., Vanderloock, A., and Baesens, B., 2013, "Process Mining for the Multi-Faceted Analysis of Business Processes—A Case Study in a Financial Services Organization," Comput. Ind., **64**(1), pp. 57–67.
[28] Rojas, E., Munoz-Gama, J., Sepúlveda, M., and Capurro, D., 2016, "Process Mining in Healthcare: A Literature Review," J. Biomed. Inf., **61**, pp. 224–236.
[29] Gunther, C. W., and van der Aalst, W. M. P., 2007, "Fuzzy Mining—Adaptive Process Simplification Based on Multi-Perspective Metrics," Fifth International Conference on Business Process Management (BPM 2007), Berlin, Sept. 24–28, pp. 328–343.
[30] Maggi, F. M., Mooij, A. J., and Van Der Aalst, W. M. P., 2011, "User-Guided Discovery of Declarative Process Models," IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011), Paris, France, Apr. 11–15, pp. 192–199.
[31] Maggi, F. M., Burattin, A., Cimitile, M., and Sperduti, A., 2013, "Online Process Discovery to Detect Concept Drifts in LTL-Based Declarative Process Models," OTM Confederated International Conferences on the Move to Meaningful Internet Systems, Graz, Austria, Sept. 9–13, pp. 94–111.
[32] Diamantini, C., Genga, L., and Potena, D., 2016, "Behavioral Process Mining for Unstructured Processes," J. Intell. Inf. Syst., **47**(1), pp. 5–32.
[33] Sinha, A., and Paradkar, A., 2010, "Use Cases to Process Specifications in Business Process Modeling Notation," IEEE International Conference on Web Services (ICWS), Miami, FL, July 5–10, pp. 473–480.
[34] Friedrich, F., Mendling, J., and Puhlmann, F., 2011, *Process Model Generation From Natural Language Text*, Springer, Berlin.
[35] Hinton, G. E., and Salakhutdinov, R., 2009, "Replicated Softmax: An Undirected Topic Model," 23rd Annual Conference on Neural Information Processing Systems (NIPS 2009), Vancouver, BC, Canada, Dec. 7–10, pp. 1607–1614.
[36] Bengio, Y., 2009, "Learning Deep Architectures for AI," Found. Trends Mach. Learn., **2**(1), pp. 1–27.
[37] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D., 2014, "The Stanford CoreNLP Natural Language Processing Toolkit," 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL), Baltimore, MD, June 22–27, pp. 55–60.
[38] Lan, L., Wu, X., and Liu, Y., 2015, "Designing a Fast Adaptive Clustering Approach for Traffic Wave Simulation," ASME Paper No. DETC2015-47873.