Pattern Discovery from Innovation Processes

Claudia Diamantini, Laura Genga, Domenico Potena, and Emanuele Storti
Dipartimento di Ingegneria dell’Informazione
Università Politecnica delle Marche
via Brecce Bianche, 60131 Ancona, Italy
{c.diamantini,l.genga,d.potena,e.storti}@univpm.it

Abstract—Innovation management and promotion has become one of the most important topics in the literature about business and executive decision support. In particular, the relationship between innovation and collaboration, both intra- and inter-organization, is gaining an increasing attention in many works, for example in the Open Innovation research field [2]. Innovation activities, especially those that involve collaboration, are typically not structured; they don’t follow a predefined scheme or procedure and are influenced by multiple factors, for instance the individual behaviour, that makes it difficult to apply classical methods of process analysis. In this paper we describe a methodology to discover significant and recurrent patterns in innovation activities, that can be used to support and improve such kind of processes. To evaluate our approach we conducted a set of experiments on a synthetic dataset, which contains a set of traces of innovation activities generated from some abstract templates, drew with the aim to model the typical ways in which innovation is carried on.

Keywords—open innovation; clustering; process mining; pattern discovery; collaboration analysis

I. INTRODUCTION

In highly dynamic environments, and especially in the current economical situation, one of the most important skills for an organization is the capability to adapt the offered services and its working mechanisms to changes. Innovation is today seen as one of the most promising ways to react in a fast and effectively way to such changes. Many different contributions, especially in economy and business management fields, recognize the importance of collaboration, both intra- and inter-organization, in order to conduct a successful innovation policy (e.g., [2]).

Innovation processes, since they refer to dynamic environments and for the inherent innovation characteristics, are not only complex and low-structured, but also various and heterogeneous in shape, content and results; they can be considered as goal-driven sets of collaborative activities, in which connections among activities emerge from organizational practices or individual behaviours. According to the terminology firstly introduced in [11], we can position innovation processes at the extreme right of the range between lasagna and spaghetti processes. While the former term delineates structured and typically organized processes, like business processes, the latter usually refers to semi- or un-structured processes in which the information flow is typically chaotic. Although fairly complex, the analysis of this kind of processes is recognized to be potentially effective for retrieving unknown patterns, which can be useful to discover valuable information for the management of innovation.

This would require management platforms capable to track activities related to innovation, so to provide support to an analysis-based innovation governance. However, at the best of our knowledge there are no proposals in the literature dealing with automatic analysis of collaborative innovation processes. This could be partially motivated by the context in which innovation itself is performed, usually among multiple heterogeneous and distributed environments (e.g., emails, social networks). Therefore, such a proliferation of platforms makes the gathering of data about innovation very challenging in the first place, and contributes to bring more complexity into play.

The only framework known to us, which is consistently oriented at giving advanced support to innovation practices in distributed and heterogeneous environments, is BIVEE\(^4\), an EU project that aims to develop a conceptual reference framework, a novel management method and a service-oriented ICT platform to promote innovation in virtual enterprise environments, i.e. real collaborating enterprises. In the context of such a project, the present contribution follows the above-mentioned approach towards the automatic support to innovation management, by introducing a process mining methodology aimed at the discovery of frequent patterns from daily innovation practices. Such patterns can be used to investigate how the innovation is performed in an organization, with both the aims of obtaining a better comprehension and improving the performance of such kind of processes. Moreover pattern discovery techniques allow to determine the types and the impact of collaboration tasks involved in innovation. For instance, if one can determine what processes actually produced some innovative ideas or solutions it is possible to discover a relationship between the presence of certain patterns of collaboration and the success (or failure) of the process, which can represent valuable knowledge for innovation managers. Pattern discovery is also useful to investigate about the behaviour of actors involved in collaborative tasks, the internal innovation processes, the possible external and internal pitfalls and so on.

\(^4\)http://www.bivee.eu
According to the approach of the BIVEE project, we assume that each (collaborative) task carried out in an innovation process is tracked and stored by organizational support systems in the form of logs. After having obtained a compact representation of event logs as process schemas, we can apply graph-mining techniques. As a matter of facts, process schemas can be easily transformed in graphs. In particular, we apply a graph-based clustering technique capable to extract the most common sub-processes from such a repository of schemas. According to their frequency and relevance, the discovered sub-processes are hierarchically arranged in a lattice structure that can be considered as a structured collection of the most common organizational innovation practices.

The rest of this work is organized as follows: in Section II we briefly survey some relevant work on pattern discovery techniques. Section III presents the proposed methodology for translating process logs to process schemas, and the usage of a clustering technique to discover frequent sub-processes from them. As a proof of concept of such a methodology, experiments with a synthetic dataset are reported in Section IV. Finally, in Section V we draw some conclusive remarks and sketch out future extensions of this work.

II. RELATED WORK

Process Mining (PM) is defined as the set of methodologies aimed at “distilling a structured process description from a set of real executions” [13]. Hence, PM analyzes process instances, namely a set of events and their execution order, and returns process schemas corresponding to these logs.

Usually PM is applied to analyze structured process, which are produced by ERP systems, Workflow Management Systems or other process-aware enterprise systems. This mining task can be exploited to support process mapping activities, conformance checking, detection of anomalous behaviours, and so forth [11]. An interesting application of PM is the analysis of instances generated by unstructured processes (also known in the Literature as spaghetti processes[11]), which are formed by activities without a predefined set of input/output, that can be recombined in various ways, and require human experience and judgement. Innovation processes considered in the present work share these characteristics. Examples of using PM to analyze spaghetti processes are discussed in [9], [6], where PM techniques have been successfully applied to software development with the aim of discovering the schema underlying the designers’ activities. In [11] the Fuzzy Miner algorithm is used to obtain the schema related to activities of an housing agency as well as to the reviewing process of journal papers; it turns also out that clustering parts of process helps in pattern discovery of spaghetti processes and, as a zoom-out functionality, in model interpretation.

This kind of techniques are generally aimed to process schema discovery; however the adoption of a single schema to model innovation processes seems to be an oversimplification of the reality. Indeed, innovation is rarely repeated in the same way and is significantly influenced by a set of factors, like human behaviour with an elevate degree of uncertain. Therefore we adopt a pattern discovery approach instead of schema discovery one, focusing on parts of the process (i.e. patterns) more than the whole process. In particular, the PM methodology we propose is based on the one introduced in [4], where a graph-based hierarchical clustering algorithm is used to discover patterns from process schemas. This methodology differs from clustering techniques that have been proposed in the Literature. As a matter of facts, classical clustering applications are aimed at enhancing the quality of discovered process schemas [5], [10], [1], while the application of clustering techniques to process schemas themselves is almost new [8]. Major differences between [8] and the proposal discussed here are: (1) in [8] the process schema is translated in vector format and then traditional agglomerative clustering techniques are used instead of exploiting graph clustering, and (2) clusters of whole processes are generated while similar substrucutures cannot be recognized.

III. METHODOLOGY

Given the availability of a logging system capable to keep track of the traces about the innovation activities carried on within a process, it is possible to represent the relations among such activities as a process schema, i.e. a workflow with a certain degree of formalism, as explained below. At each stage of the process, several similar and alternative activities can be used for every specific need. For instance, both a physical meeting, where people meet together in an office, and a virtual meeting through Skype or other communication technologies, carry out the same function and therefore can be considered, at a certain level of abstraction, as alternative instances of a generic class of “meeting” activities. According to such considerations, in this work we assume to have at disposal a classification of the activities in classes and their instances, through a taxonomy that is to be compiled by collaborating enterprises. These are actual hypotheses in the context of the BIVEE project, one of whose main outcomes will be indeed a set of functionalities for managing, retrieving, analyzing information about previous innovation processes, in the fashion introduced herein.

The methodology we followed to achieve the goal of innovation patterns discovery is explained in the following Sub-sections and begins with the collection of the traces and their preprocessing in order to get a more concise representation of them as process schemas. Then, given that schemas have an intrinsic graph structure, it is possible to reduce the problem of Process Schema Mining to Graph Mining, by exploiting already existing graph-based techniques. This step implies the translation of process schemas into graphs and the consequent application of a graph mining clustering technique, capable to extract common and frequent sub-graphs. Such patterns can indeed be backwards rendered as schemas, which identify common patterns in the original set of innovation processes. Activities in the schemas and nodes in the graphs, exactly like the items in the log traces, can be represented through two level of abstraction: a concrete level that corresponds to the original activity’s name, and an abstract level that
corresponds to the name of its class. As a consequence of the taxonomical classification of activities, schemas and graphs can have a double representation: activity-level and class-level, whose differences w.r.t. performances in pattern discovery will be discussed in Section IV.

A. From traces to schemas

Given the availability of event logs about a specific innovation process, stored in a proper format, this phase is aimed to construct a schema model. The only assumptions made about such logs involve some mandatory information that, as in typical workflow management systems, are used to annotate the logs. In particular we refer to a minimal set of attributes, namely a numeric id or the activity timestamp, the activity name, the id of its predecessor, and the id of the specific innovation process. Other information are optional and could be useful to enrich the schema. Then, preprocessing of such traces is needed to obtain a structured and a more synthetic representation of each set of logs, in a process description language. This means to retrieve, through ad-hoc procedures or process mining techniques, parallel branches starting from the same activity of the same process, and repeated sequences of activities, that are represented as cycles.

Common process description languages allow to represent activities, sequences of activities (SEQ), parallelization (SPLIT) and merging (JOIN) operators. More precisely, SPLIT and JOIN operators are characterized by the typology of flow synchronization: a SPLIT-AND means that the end of an activity starts all the linked activities, while in a SPLIT-XOR only one will be executed. Symmetrically, a JOIN-AND indicates that an activity begins when all the previous activities are terminated, while in a JOIN-XOR the completion of a single activity is needed. In this work we refer to schemas inferred from actually executed traces, and each trace refers to a different process. Hence, parallel branches are represented only by SPLIT-AND and JOIN-AND, while cycles are represented through a JOIN-XOR operator with two alternative paths, one bringing back to the beginning of the cycle, while the other exiting the cycle. An example made of two process schemas is shown in Figure 1.a. Note that this phase is agnostic w.r.t. the specific language and format for the schemas, under the condition to refer to an enough expressive language. Many existing process mining techniques allow the preprocessing of traces and the definition of schemas [12].

B. From schemas to graphs

In order to apply graph mining techniques to extract relevant patterns, it is needed to translate the richer process schemas representation to simpler directed graphs. As we discussed in [4], the choice of the detail level for representing the process is critical as it affects the significance of the discovered knowledge, and in fact we proposed three alternative representation models, characterized by an increasing level of compactness of the corresponding graph. According to the terminology introduced in [4], in this work we refer to the C model, a compact representation in which activities are translated as nodes, while operators are omitted. Hence, SPLITs/JOINs are intrinsically rendered as multiple edges leaving/entering a node. In order to represent cycles, that were not treated in [4], here we add to such a model also an explicit SPLIT-XOR operator node. Arcs are labelled to maintain information about type of operator (“s”, “j” and “seq” for SPLIT, JOIN and SEQ respectively) and the two linked nodes. This representation is necessary to guarantee the correct interpretation of the final discovered patterns after the compression performed by the algorithm described in the next Subsection. The example in Figure 1.b shows the translation of the two process schemas of Figure 1.a into graph representation.

C. Hierarchical Clustering of graphs

Hierarchical clustering techniques are aimed at extracting frequent substructures (i.e., sub-graphs) out of a set of input graphs. Such clusters are then arranged in a hierarchy of clusters with different levels of abstraction: while the top-level clusters are defined only through elements belonging to input graphs (i.e., nodes and arcs), lower-level clusters extend upper-level clusters with other elements, implicitly defining a lattice structure. Therefore, descending the hierarchy, we pass from structures that are very common in input graphs (i.e., frequently occurring, with a high support) to structures specific for each input graph (i.e., with low support).

An example is shown in Figure 1.c where a lattice is generated by a repository of graphs, two of which are shown in Figure 1.b. Defining by $A$, $B$, $C$ specific activities and by $S_1$ substructures, graphs belonging to cluster $C_1$ contain the substructure $S_1 = \{A \rightarrow C\}$, while graphs belonging to $C_2$ contain the substructure $S_2 = \{B \rightarrow A; B \rightarrow C\}$. The cluster $C_3$ is described by $S_1$ plus an additional node $B$. Therefore, $C_3$ containing the structure $\{A \rightarrow C \rightarrow B\}$ and is a specialization of $C_1$, because the former extends the latter. Finally, $C_4$ is child of both $C_1$ and $C_2$, since it is the set of graphs where $S_1$ is linked to $S_2$. Note that different clusters can be children of the same parents: for instance, a cluster described by the sub-graph $\{S_1 \rightarrow A \rightarrow S_2\}$ would be sibling of $C_4$.

Among the hierarchical clustering algorithms, in this work we refer to SUBDUE [7] that, by iteratively analyzing input graphs, is capable to extract at each step all existing substructures and to discover the one that best compresses the graphs. After each iteration, such a substructure is then actually used to compress the graphs, by substituting a single node to each occurrence of the substructure. Hence, the chosen substructure becomes a cluster of the lattice, and the compressed graphs are presented to SUBDUE again, in order to repeat these steps until no more compression is possible.

The search for the best substructure is driven by the MDL criterion (Minimum Description Length), aimed to the minimization of the description length of the graph after the compression, i.e., the number of bits needed to represent its adjacency matrix. Such an index globally takes into account both the dimension of a substructure and its frequency. In order to discover substructures, isomorphic graph matching is performed, although SUBDUE can implement also inex-
act matching. Furthermore, some computational-constrained heuristics are introduced to reduce the search space. The algorithm has been successfully applied to analyze structured objects in several domains (see http://ailab.wsu.edu/subdue/).

D. Lattice Evaluation

Being a hierarchical clustering lattice a clustering model, its quality should be assessed by evaluating the intra-cluster homogeneity of each cluster and the inter-cluster heterogeneity of different clusters. A good model is characterized by both high homogeneity and high heterogeneity, that is any clusters is representative of very similar graphs and few (or no) overlaps among different clusters exist. Indeed, due to its hierarchical nature, a lattice is characterized by a high overlap among all children of the same cluster. Hence, we can not use well known clustering evaluation measures, namely intra- and inter-clusters measures, but we have to refer to measures that take into account the structure of the hierarchy as well. In [3], [7] some measures are introduced to evaluate the lattice discovered by SUBDUE. In particular [7] proposes and discusses the diversity (DIV) as a heterogeneity measure, which is based on evaluating the number of manipulations (i.e. insertions, deletions and updates) needed to transform a cluster into another. The diversity is computed at every level of the hierarchy, hence weighing several times the diversity of a cluster with more than one parent. During the lattice generation, SUBDUE may discard those substructures having very low compression capability, loosing some input nodes or arcs. Hence, both [3] and [7] introduced a measure evaluating the degree of coverage of the model. This measure, named completeness, checks the number of original graph elements still present in the final lattice. In order to evaluate a clustering model, it is also important to measure the cardinality of a cluster, representing the number of times the related substructure occurs in the input graph. To this end we refer to two measures: frequency and representativeness (REP). The main difference between the two measures is that the latter does not consider repetitions of substructures in a graph, measuring the number of input graphs holding the given substructure at least once. For example, given a simple dataset of two graphs $G=\{(A\rightarrow C); (A\rightarrow C\rightarrow A\rightarrow C)\}$ and the lattice in Figure 1.c, the REP of the cluster $C_1$ is 2 and its frequency is 3. Both these measures have to be taken in proper account when the hierarchical cluster is used for information retrieval tasks.

IV. EXPERIMENTS

In this section we discuss the application of the methodology to a set of innovation process traces. Unfortunately, due to the limited diffusion of innovation management systems easing the collection of activity traces, combined with the general attitude of enterprises to protect sensitive information related to innovation, real-world data are not available. Hence, in order to test the approach, we generated a synthetic set of process traces. The following Subsection describes the methodology adopted to generate the dataset and its characteristics, as well as the experimental setting. In subsection IV-B we present and discuss the generated lattice.

A. Experimental Setup

In order to generate the set of traces for the experiments we started from general innovation templates, from which it was possible to produce a desired number of traces and hence process schemas. We defined such templates using both our personal knowledge of innovation processes and the Literature about such kind of activities; in particular we focused on works regarding the correlation between collaboration and innovation [2]. Moreover, we tried to take in account also the different contexts and modalities in which collaboration can take place, with the aim of identifying some principal steps that are in common between different collaborative paths. The final result of this work consists of three abstract templates, shown in Figure 2, each representing a different innovation model.

The Template 1 describes the evolution of an innovation project, where a team (possibly made of a single person) discovers some kind of opportunity and, after a first phase of internal development, involves external experts in its activities. It should be noted that we have taken as example an environment where the “opportunity” is represented by the discovery...
of a financing opportunity, or “call”, but obviously the model can cover a broader range of market pull opportunities. If at least one of the external partners accepts to join the project, the scheme proceeds with a meeting activity, that can require further steps of parallel individual tasks. Individual contributions are then merged together and the final ideaRefinement activity produces a proposal ready to be developed. Branches and cycles are included: for instance, after the final ideaRefinement activity the process can move to the end or come back to the meeting activity, repeating the sub path between these two steps for an arbitrary number of times. Finally, in order to model the possibility that the collaboration fails, we introduced an end point in the template which follows meeting activities, to allow the process to terminate without producing the final idea. Similar end points are introduced also in other templates. The second template is inspired by an open innovation model, where an organization decides to involve external partners since the idea definition. In this case the enterprise identifies a particular goal to reach and, instead of starting the project internally, invites some external members to submit their ideas or proposals aimed to reach its objective. Submitted ideas are filtered by an internal evaluation team before or after a voting procedure, where each idea receives a certain number of votes and those with the greatest number of votes are
the winners. The way internal team evaluation and voting procedures are ordered determines two different sub-templates, namely Template 2-a and Template 2-b in Figure 2.

Finally the Template 3 represents a simple example of “client-pull” innovation in traditional enterprises; the process starts from a particular client request and proceeds with a set of activities where it is supposed that the client and the enterprise strictly collaborate together. In particular every solution proposed by the enterprise has to be approved by the client before the process can continue.

It should be noted that the templates we built do not have the intent to be exhaustive of all possible paths of collaboration that can be found in real innovation activities; they want to be simply some relevant examples, that can be used to derive reasonable traces for our analysis. Clearly one can divide and recombine the templates, obtaining many other different paths and situations, but this kind of analysis goes beyond the goal of the present work.

Starting from the templates we generated the input dataset for SUBDUE. Traces are obtained by randomly setting the number of parallel branches, the number of iterations, the alternative branch followed and so forth. As an example, let’s consider the first template. First of all we can observe that a process can involve just one callDiscovery and hence directly moves to the ideaRefinement step; otherwise we can find more callDiscovery activities, that are joined in a meeting activity. After the latter, the process can terminate or proceed. The number of callDiscovery can vary in a range from a minimum of one to a maximum of five, and this range is fixed also for the others parallel branches. The next step consists of one or more collaborationProposal: each proposal is followed by a collaborationAnswer that can represent a rejection, (i.e., the branch terminates), an acceptance (the branch can proceed), or a negotiation (several repetitions are possible until it stops or an agreement is reached). If there is at least one branch with a positive answer, the process moves to the second meeting; from here, it can either terminate, proceed to the final ideaRefinement, or pass through individualContribution activities. Note that in this case the minimum number of parallel execution is fixed to two, because otherwise there could not be a job distribution. Finally, this sequence can be repeated an arbitrarily number of times. Each box in the templates represents a class of activity, which in order to generate the input dataset, assumes values in a predefined set of instances.

We proceeded in a similar way also for the other two templates, setting for each of them specific random values for each point of variability: as a result, after the transformation phase introduced in III, we obtain a dataset composed of 200 schemas (i.e., graphs). Table I shows some statistics about the dataset, while in Figure 3 the dataset is graphically depicted in order to enlighten its spaghetti-like nature.

We conducted two different experiments, the first one by considering the activities and the second the classes by substituting to activities the first-level class of the taxonomy. In both experiments the hierarchical clustering has been obtained by running the SUBDUE algorithm with the parameter beam, which affects the number of substructures to consider at each iteration of the algorithm, set to 4, 10 and 20. In order to improve the execution time, SUBDUE has been executed in its pruned version, where at each iteration a substructure (and the sub-graph of its children) is not considered if its compression ratio is less than the value assigned to its parent. The other parameters have been maintained to their default values. Table II shows the results we obtained for both the experiments.

<table>
<thead>
<tr>
<th># graphs</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td># graphs T1</td>
<td>34</td>
</tr>
<tr>
<td># graphs T2-a</td>
<td>50</td>
</tr>
<tr>
<td># graphs T2-b</td>
<td>46</td>
</tr>
<tr>
<td># graphs T3</td>
<td>70</td>
</tr>
<tr>
<td>nodes</td>
<td>1973</td>
</tr>
<tr>
<td>arcs</td>
<td>2123</td>
</tr>
<tr>
<td>min # nodes/graph</td>
<td>2</td>
</tr>
<tr>
<td>max # nodes/graph</td>
<td>33</td>
</tr>
<tr>
<td>avg # nodes/graph</td>
<td>9.89</td>
</tr>
<tr>
<td># unique labels</td>
<td>51</td>
</tr>
<tr>
<td># unique class labels</td>
<td>17</td>
</tr>
</tbody>
</table>

B. Discussion of Results

From Table II it turns out that the experiment with classes demonstrates the best average performance. As a matter of facts, the lattice is around six-times more compact, and it has higher completeness and representativeness of first top substructures. When instances are considered, the value of diversity is higher than experiments with classes; this is mainly due to the increased variability of the labels of nodes and arcs (51 vs 17). Since the structure of discovered lattice, and in particular its top level substructures, does not significantly change varying the beam value, in the following we refer to results obtained with the beam set to 4.

Figure 4 zooms on the first 5 top-level substructures of the discovered lattice, and some of their children. Substructures are labeled according to the iteration that has generated it, and those with the higher compression capability are discovered first. Recall that compression capability is determined by both the frequency and dimension of a substructure.

Analyzing patterns in figure 4 we observe that:

- SUB_1 represents a collaborative pattern typical of the open innovation modality with ideaSubmission followed by two parallel voting activities, followed by voting collection. It is interesting to note that this structure, which is the best compressing one, contains two voting activities and is the only top level substructure containing voting. Hence, each process where voting is performed has at least 2 voting activities: voting is indeed a collaborative activity which cannot be performed by a single internal group. In other words, the pattern describes a characteristics of the organizational structure and is not
limited to the description of the flow of activities, as it would be if just the sequence ideaSubmission → voting → votingCollection was discovered.

- **SUB_2** represents the alternative way of evaluating a submitted idea contained in Template 2. It is performed by a single person or a group of people in synchronous. The fact that ideaAdmission is followed by the end node enlightens that most of the proposed ideas are discarded in our dataset. Although SUB_1 appears in less than half processes with respect to SUB_2 (20 versus 42) the frequency of appearance of SUB_1 is 90, while it is 105 for SUB_2. These frequencies, together with the fact that SUB_1 contains a higher number of nodes and arcs, accounts for the order of discovery of the two patterns, but shows that the method extracts as typical patterns those substructures that are maximally frequent independently of the frequency of the processes: the launch of one single collaborative voting initiative on the internet producing a large number of responses could mask the typical attitude of an enterprise to evaluate ideas internally. Hence the method should not be interpreted as a way to recognize the typical innovation practices as a whole, but only significant fragments of them.

- Figure 4 also shows SUB_9, one of the children of SUB_1. SUB_9 does not add much information with respect to SUB_1, it simply states that three parallel voting activities are also frequent. This can be considered irrelevant for the analysis of innovation activities patterns, unless we are actually interested in understanding the number of involved voting partners. On the other hand, looking at SUB_6, child of SUB_2, we can appreciate the mechanism of call for ideas that precedes the ideaSubmission. So we could not a-priory discard the lower lattice levels. The complexity of the produced lattice, containing this kind of details, can be considered as a limit of the approach.

- **SUB_3** enlighten the third way by which an idea is submitted, which is typical of the “client-pull” modality of the Template 3.

- The modality related to opportunity-pull of the Template 1 is shown in SUB_4. It is also interesting to note that its children SUB_10 contains four parallel callDiscovery → ideaSubmission followed by a meeting, meaning that parallel activities of opportunity analysis performed by independent groups is very frequent in the dataset. This can be interpreted as an inefficiency due to poor organization and coordination, or it can be recognized as a planned and desired pattern targeted at increasing the variety of ideas and the probability of success. In any case, it represents a valuable discovered knowledge for innovation managers.

- Finally, SUB_5 presents another typical pattern of the opportunity-pull template, related to the search for partners phase. The fact that collaboration proposals followed by collaboration answer turns out to be among the most frequent substructures, enlighten that a (maybe optimistic) hypothesis has been made in dataset construction, namely that most of the times people kindly give an acknowledge to the request. However these feedbacks often do not correspond to an acceptance of the request, otherwise activities following collaborationAnswer would appear in the structure as well.

Summarizing, from the analysis of discovered substructures, we can appreciate both significant structural and organizational characteristics of the designed template, as well as statistical properties of the generated instances.
V. CONCLUSION

The paper discussed the adoption of a graph-clustering technique for the discovery of typical innovation activity patterns. The use of graph-clustering is original with respect to the state of the art application of process mining to business process analysis in general, and represents a pioneeristic approach to innovation process analysis. The experimentation performed on synthetic data shows the capability of the method to recognize the most important patterns included in innovation activity templates, providing valuable information about both structural and organizational characteristics, as well as statistical properties of the generated instances, demonstrating itself as a promising approach for an organization oriented analysis of innovation. On the other hand, some limits has been observed as well, mainly related to the presence of unnecessary details and to the complexity of the generated lattice. In the future we plan to extend the experimentation to real innovation processes and to study different representation alternatives to deal with the limits observed and to discover different kinds of information. For instance, representing parallel identical flows only once with a cardinality information would allow to simplify the discovered structure and to better recognize activity flows, when this perspective is considered relevant. We also plan to compare our approach to other process mining techniques usually adopted for unstructured process analysis.

REFERENCES


