A Genetic Algorithm for the Inductive Derivation of Reference Models Using Minimal Graph-Edit Distance Applied to Real-World Business Process Data

M.Sc. Alexander Martens, Privatdozent Dr. Peter Fettke, Prof. Dr. Peter Loos

Institut für Wirtschaftsinformatik im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI GmbH) und Universität des Saarlandes, 66123 Saarbrücken, E-Mail: alexander.martens@iwi.dfki.de

Abstract

Business process management has become an important topic and a subject of lively discussion, especially the conceptual modeling of business processes. This task is time-consuming and the outcome depends strongly on the mindset of the designer. Meanwhile, recent research is focusing on inductive reference modeling, especially based on minimal graph-edit distance. In contrast to related work, this innovative approach operationalizes reference models as an abstract model that can be transformed in a minimal number of steps towards individual business process models. The formulated optimization problem of minimal graph-edit distance is approximated using a variant of genetic algorithms. The method is applied to relevant real-world examples of individual business process models to evaluate inductive reference model development following evolution strategies. Therefore, the presented approach is implemented prototypically as proof-of-concept, and thus proves its practical usage.

1 Introduction

1.1 Motivation

Business Process Management (BPM) has become an important topic and a subject of lively discussion, especially the conceptual modeling of business processes. BPM provides important concepts, methods and techniques for the organizational practice, e.g. different languages for business process modeling. In general, individual models are developed to document and analyze business processes, e.g. for the purpose of process optimization [19]. The modeling task is time-consuming and the realized value of the result depends strongly on the mindset of the process designer. Among other reasons, that is why the trend within organizations is going into the direction of reusing existing practices as a reference for the design process in order to save costs and to assure a higher quality of one's own individual business process models [3]. Therefore, reference models are an important topic that is still being discussed and researched.

Reference modeling approaches can either follow the deductive or inductive development strategy [12]. While deductive development is based on general theories, the inductive modeling is based on consolidation of real-world business process data. Beside the described use case of optimizing the modeling process within one organization by having a reference model at hand, the inductive reference model development might be of great use in other current problem fields as well. The harmonization of business processes that are typically behind of individual business process models, evolving as output of controlled modeling from different perspectives and point of views of different employees for example is one interesting use case. Another one is the analysis of discrepancies between different business processes within the same application domain, which is also supported by reference model mining. So far, research has mainly focused on the deductive strategy, and only partially on the inductive one. Furthermore, no consistent definition of the reference model terminology exists, although a common understanding in the reference modeling community has developed into the direction of universal applicability and reusability as most important characteristics [2], [12], [28].

- Universal applicability: A reference model has to be valid for as many organizations as possible within the same domain, which were not necessarily considered during the inductive development process.
- Reusability: A reference model has to be adapted to individual models describing the same business process in as few modeling steps as possible.

These characteristics lead to positive economic benefits [2], increasing modeling effectiveness and efficiency, resulting in cost and time reduction [3]. Motivated by the unused potential of the inductive strategy, the relevance for a systematic method to operationalize reference models is given. This fact is addressed by the following work that can be classified as design science research [18].

1.2 Problem statement

Given individual business process models as input, each model is considered as directed graph. The input data is required to consist of several real-used and proven business processes of satisfying quality, modeled in the same modeling language, preferably with SESE regions to simplify recombination. It is assumed that business process models add up to a representative selection, including as many variants as available of business processes out of the same application domain in order to represent profound data. For example, the individual models can be acquired by professional service firms (e.g. strategic consulting companies) among different organizations. The presented method integrates over this set of individuals in order to find a reference model similar to the variants by definition, using a genetic algorithm for solving the classical NP-hard [14] optimization problem of minimal graph-edit distance (GED) between the reference model and all individual models, proven by the reduction to the travelling salesman problem. No special optimization procedure exists, so that a general method has to be applied to obtain an approximated solution. This strategy of finding an abstract model is opposed to the integration of the individual models into some kind of super-model. The minimal graph edit distance [13] is defined as the minimal number of operations to transform one graph into another. Valid operations are the insertion, substitution and deletion of nodes and edges. In general, the problem belongs to the class of NP-complete problems [31] due to the underlying mapping problem. Regarding this problem, it is assumed in this paper that the mapping between the nodes of two different graphs is given a priori. Identical nodes are identified by identical labels or defined as gold standard, provided for example in form of a map file to the algorithm. Evolutionary algorithms [16] are initialized by an initial population of individuals in a first generation. A genetic algorithm belongs to the classical variants of evolutionary algorithms, being the closest ones to the paradigm of the biological evolution in the sense of terminology and methodology. It provides a framework for solving optimization problems. The variant described in this paper is designed for the application to real-world business process data.

This paper is organized as follows: After a survey of recent related work (section 2), the calculation of graph-edit distance as fitness function is introduced (section 3). Then, an overview of the entire optimization process is presented, describing each component of the genetic algorithm in detail (section 4). Afterwards, the results of the method are shown and evaluated (section 5), applied to real-world examples of business processes. Furthermore, the overall approach is discussed (section 6) and findings are concluded based on the results (section 7). Finally, future work is exposed, addressing the current limitations of the method in relation to the requirements and assumptions (section 8).

2 Related work

In recent research, especially the graph-edit distance has been applied to inductive reference model development approaches. Ardalani et al. [1] demonstrate a heuristic method, defining a minimal cost of change function based on minimal graph-edit distance. The reference model is iteratively developed by using this function in order to match the underlying individual business process models. Compared to our approach, we have formulated the problem as optimization problem, approximated by the powerful concept of genetic algorithms, not limiting the possible solution in the way it is constructed. In general, genetic algorithms are easy to extend and generally applicable without the need of making assumptions about the optimization problem itself. That is the reason why more and more genetic algorithms are considered for modeling and optimization purposes. For example Genetic Process Mining (GPM) tries to derive processes based on logs [7]. Chang et al. [6] and Gen et al. [15] apply genetic algorithms for project management purposes to solve resource planning, network model and optimization problems. Yahya et al. [30] have used first genetic algorithms for the inductive derivation of reference models. However, the fitness function is defined by proximity distance measure based on incoming and outgoing edges of nodes. The proposal in this paper, making use of GED to measure the similarity, is more evident as operationalization. Also, the results suffer from the fact that no business process language for modeling is used. The application to real-world examples and the evaluation thereof are missing.

Instead of using genetic algorithms, related work tries to generate reference models based on other heuristic methods and metrics. Li et al. [23] present a cluster-driven approach that encompasses a reference model out of business process variants, minimizing its average distance to the variants. Necessary preconditions restrict the application to block-structured models. Simulated-annealing approaches (cf. [26], [29]) divide the individual models first into isomorphic sub-graphs. Subsequently, relative frequencies are calculated for each sub-graph to determine the sub-graphs together with an abstraction parameter, thus incorporating the sub-graphs into the reference model. In contrast to that, genetic algorithms maintain a pool of solutions rather than one. Other recent research activities are going into the direction of generic algorithms for automatic merging of two business process models in order to build a union, comprising the behavior of both [22]. The instrumentation of such generic algorithms allows abstraction-based inductive reference model development using elimination and aggregation as operations upfront (cf. [25]), which is not required by the application of our method.

In the research area of pattern recognition, the introduction of finding a graph that is a pattern for a set of others by means of a genetic algorithm is known as median graph computation problem. The concept is introduced in [20] based on graph-edit distance as similarity measure, but defined differently in comparison to our definition (cf. section 3). This work brings first together genetic algorithms with GED, but applied in different variations and research areas. In [10], [24] the problem is motivated from biological imaging application and other biomedical applications. Nevertheless, the concept is inspiring this work, but adapted to the application to real-world business process models as another variant, dealing with their characteristics in order to find a reference model. Apart from different problem fields, the calculation of median graphs varies especially in applied optimization methods. A reduction of complexity is tried to be achieved in [11] by embedding graphs in vector space instead of using genetic algorithms, where the median graph is computed based on graph-edit distance, applied to biological image analysis. The question for the right optimization method is supported by a comparison of genetic algorithms and a combinatorial solution [5].

Overall, this presented algorithm differs from existing methods in different aspects regarding design of genetic operations, graph-edit-distance calculation, initial population initialization and internal graph representation, even though the concept is not a wholly new idea. The novelty of this paper lies more in the application and evaluation of genetic algorithms as global optimization solver for the optimization problem of minimal graph-edit distance in the sense of inductive reference model operationalization as an important part of conceptual business process modeling. The specifics are identified and discussed. As proof-of-concept, the application of the prototypical implementation to individual business process models is validated with the aim of determining a valid reference model. Thus, the practical usage is demonstrated.

3 Graph-edit distance

Business process models like event-driven process chains (EPCs) are representable as directed graph structures, underlying certain syntactic rules. In general, a graph consists of a set of nodes and a set of edges, each edge connecting two nodes. In the context of business process models, different types of nodes and edges exist, e.g. in EPCs a node represents an event or function, implicating a different meaning. A widely-accepted similarity measure for graph comparison is the minimal GED [4]. The similarity is defined by the number of edit operations necessary to transform one graph into another. Insertion, deletion as well as substitution of nodes and edges are possible edit operations [8]. Combinations of these edit operations allow the splitting of a node into two nodes and the combination of two nodes into a single one. The calculation of GED requires a mapping between nodes and edges in both graphs. In a general graph, the problem of finding such a mapping, minimizing the GED, is NP-complete. But independent of the used modeling language, business process models describe a process flow, represented as directed graph, inducing a pre-defined order of nodes. Starting at the root nodes of two process models, subsequent nodes are successively integrated extending breadth-first-search (BFS). It ends up with a heuristic one-to-one mapping, based on the plausibility that nodes closer to the start of a process should be mapped to nodes closer to the start of the other process. A reasonable trade-off between complexity and precision is in the main focus. Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, this greedy algorithm allows us to find an approximate solution for this mapping problem under a reasonable trade-off between complexity and precision. The method is separated into two phases. First, the nodes that occur only once in a business process model are considered. Afterwards, the nodes occurring multiple times complete the mapping as possible. For example, in many EPC models the same operator is used multiple times and events and functions can occur in different control flow branches. In equation 1 the node mapping function is specified.

$$\mathrm{map}(v_1) = \begin{matrix} v_2 & \text{if } \mathrm{label}(v_1) = \mathrm{label}(v_2), v_1 \in V_1, v_2 \in V_2 \\ \text{undefined} & \text{else} \end{matrix} \qquad (1)$$

Finally, the mapping of an edge $(u, v) \in E_1$ is established, if either $\mathrm{map}(u)$ or $\mathrm{map}(v)$ are undefined or if $(\mathrm{map}(u), \mathrm{map}(v)) \in E_2$. Unmapped nodes and edges are regarded as inserted or deleted, because they appear only in one of both graphs. In case a node is neither inserted nor deleted, it is substituted [8] and the incident edges are preserved. This concludes that no edit operation is necessary, as defined by term *subv*, under the condition that the mapped nodes are identical. Let *sn* be the set of inserted or deleted nodes, *se* be the number of inserted or deleted edges and let *sim* be a function that assigns a similarity score to a pair of nodes. Then, the GED similarity measure (see equation 2) is defined as follows (cf. [9]), gained directly from the number of graph edit operations that are needed to transform one graph into another, given the underlying mapping of nodes and edges a priori.

$$\mathrm{GED}_{\mathrm{sim}} = 1 - \frac{\mathrm{snv} + \mathrm{sev} + \mathrm{subv}}{3} \tag{2}$$

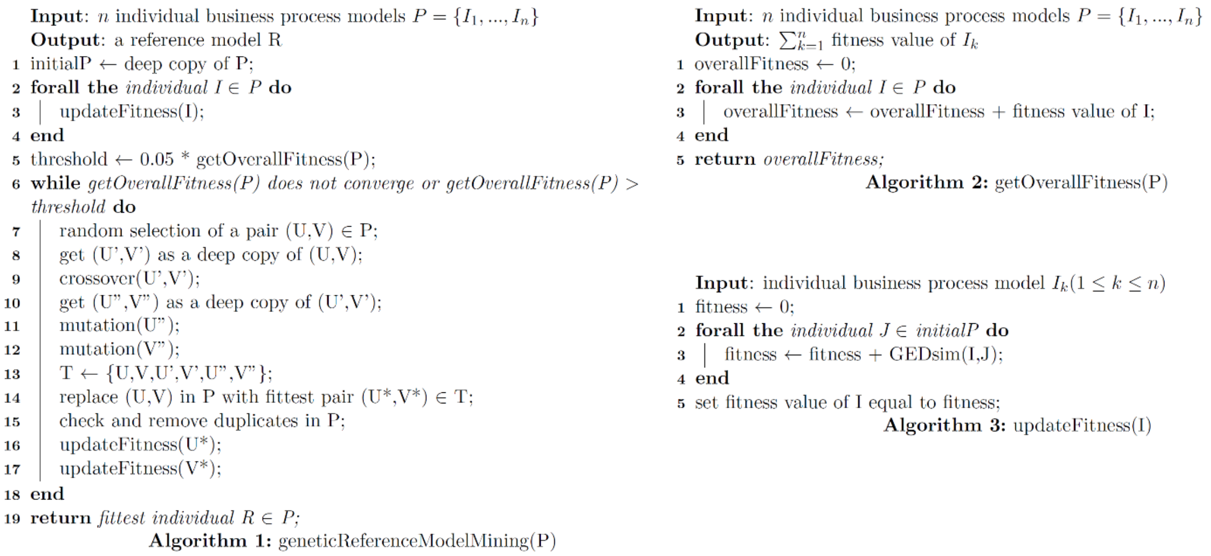
$$\mathrm{snv} = \frac{|\mathrm{sn}|}{|\mathrm{V}_1| + |\mathrm{V}_2|} \tag{3}$$

$$\mathrm{sev} = \frac{|\mathrm{se}|}{|\mathrm{E}_1| + |\mathrm{E}_2|} \tag{4}$$

$$\mathrm{subv} = \frac{2\sum_{\mathrm{v}_1 \in \mathrm{V}_1, \mathrm{map}(\mathrm{v}_1) = v_2} 1 - \mathrm{sim}(\mathrm{v}_1, \mathrm{map}(\mathrm{v}_1))}{|\mathrm{V}_1| + |\mathrm{V}_2| - |\mathrm{sn}|} \tag{5}$$

4 Genetic algorithm

Evolutionary algorithms [16] are a class of stochastic-heuristic optimization methods that are based on principles of biological evolution. Instead of problem-specific heuristics, these algorithms are universal and the flow of single steps can be applied to many different kinds of problems. An important advantage in comparison to traditional optimization methods is that evolutionary algorithms do not make any assumptions about the optimization problem itself, resulting in better approximated solutions. The type of evolutionary algorithms closest in terms of terminology and methodology to biological evolution are genetic algorithms. Evolutionary algorithms are able to solve also non-linear or discontinuous problems, but the computing time is comparatively high and unpredictable.



```
Input: $n$ individual business process models $P = \{I_1, ..., I_n\}$
Output: a reference model R
initialP ← deep copy of P;
forall the individual $I \in P$ do
    updateFitness(I);
end
threshold ← 0.05 * getOverallFitness(P);
while getOverallFitness(P) does not converge or getOverallFitness(P) > threshold do
    random selection of a pair (U,V) ∈ P;
    get (U',V') as a deep copy of (U,V);
    crossover(U',V');
    get (U",V") as a deep copy of (U',V');
    mutation(U");
    mutation(V");
    T ← {U,V,U',V',U",V"};
    replace (U,V) in P with fittest pair (U*,V*) ∈ T;
    check and remove duplicates in P;
    updateFitness(U*);
    updateFitness(V*);
end
return fittest individual $R \in P$;
```

Algorithm 1: geneticReferenceModelMining(P)

```
Input: $n$ individual business process models $P = \{I_1, ..., I_n\}$
Output: $\sum_{k=1}^{n}$ fitness value of $I_k$
overallFitness ← 0;
forall the individual $I \in P$ do
    overallFitness ← overallFitness + fitness value of I;
end
return overallFitness;
```

Algorithm 2: getOverallFitness(P)

```
Input: individual business process model $I_k (1 \leq k \leq n)$
fitness ← 0;
forall the individual $J \in initialP$ do
    fitness ← fitness + GEDsim(I,J);
end
set fitness value of I equal to fitness;
```

Algorithm 3: updateFitness(I)

Figure 1: Genetic algorithm for inductive reference model derivation in pseudo-code

The focus regarding the variant design of genetic algorithms lies mainly on the internal representation of individuals, the initialization of the first generation, the specifically tailored functionality of genetic operations and the selection criterion as well as selection process of individuals from generation to generation. Given a set P of individual business process models as input, the described genetic algorithm in figure 1 integrates over the individual models in order to find inductively a reference model, minimizing the overall graph-edit distance. The function that has to be minimized is called fitness function, assigning to each individual a fitness value. The individual business process models represent the initial population of size n, building up the first generation as possible reference model candidates that are evolved during the algorithm to better solutions. Beginning with the first generation, in each iteration a pair (X, Y) ∈ P is randomly selected and the genetic operations crossover and mutation are applied sequentially. The random function is influenced by the fitness values of every individual in the sense that the probability for selecting a specific individual decreases quadratically from strongest to weakest fitness, showing a positive effect on the results. The crossover operation determines randomly one node in graph X and Y out of single-entry single-exit regions (SESE). SESE [21] is defined for a node x when:

- node x dominates a node y in a directed graph (an ordered edge is connecting x and y) if every path from start to y includes x.
- node y post-dominates a node x in a directed graph (an ordered edge is connecting y and x) if every path from y to end includes x.
- every cycle containing x also contains y and vice versa.

SESE fragments representations of business process models (e.g. EPCs) into connected, control-independent sub-graphs with one control flow entry as well as one control flow exit without back-edges, entering or exiting the region. In general, nodes which are start or end nodes of SESE regions are unique points for splitting graphs in multiple sub-graphs in order to exchange those parts between graphs during the crossover operation. Here, the graphs X and Y are split in an upper and a lower part at such two points. The tail of X^{upper} is connected to head of Y^{lower} and the tail of Y^{upper} is connected to head of X^{lower}. In the case that no SESE fragments can be identified depending on the business process modeling language, the identification of the first connected component starting BFS downwards at the randomly selected node enables also the splitting into two independent graph components with discrete set of nodes and edges in the end. But, it is necessary to define and follow a set of rules, which is increasing the complexity of recombination. Given a mutation probability, the mutation operation is applied or not. During mutation process, two nodes are randomly defined and exchanged, or one node is deleted randomly. The mutation fulfills the purpose of overcoming local optima in order to find a global one. Both operations are designed independently from a concrete modeling language by definition on the graph structure as abstraction layer, but syntactical correctness of the outcome is assured by general rules, valid for all modeling languages. Furthermore, these probabilistic operations allow the growth of new individuals forming the next generation, but only the fittest ones survive. This process is repeated until the termination criteria are fulfilled. The solution of the optimization problem is found in the fittest individual within the final generation that is defined as the individual with the minimal graph-edit distance to all other individuals.

The designed algorithm (see figure 1) has to fulfill the following requirements:

- Correctness: In economics, many different kinds of business process modeling languages exist. A very common example, where this paper is exemplary based on, are EPCs that are representable as directed graphs with different node types – events followed by functions. In addition, logical operators like AND, OR or XOR are available, connecting two branches of a business process. Based on this example, it is evident that syntactical correctness is important and has to be satisfied for resulting reference models. The correctness is not destroyed whatever the different modeling languages of the different business process models as input of the algorithm, because the structural order and flow of nodes of different types are preserved by applying operations only to nodes of the same type. The deletion of nodes only takes place if there is a clear predecessor and successor node. In this exceptional case, the structural order of different node types can be destroyed, but the automatic correction is done in a post-processing step in dependency of the modeling language.

- Efficiency: The complexity of real business process models is generally high due to a large number of entities, having a strong impact on runtime and memory consumption. In computer science, different data structures for representing graphs are possible. These structures are different in runtime complexity regarding the implementation. The efficient handling of the genetic operations requires a data structure that enables efficient structural graph transformations and the storage of business process model metadata information, e.g. the type of a node. In our implementation, a pointer structure of modeled objects is used instead of a simple adjacency matrix.

5 Evaluation

5.1 Software prototype

The described algorithm is prototypically implemented in the Java programming language and integrated in a self-developed application within our research group, providing the functionality of loading business process models that are represented in a common XML-based interchange format as for example EPC Markup Language (EPML) or ARIS Markup Language (AML). For the implementation of the algorithm itself, only the JAMA library (http://math.nist.gov/javanumerics/jama) is re-used for calculative purposes. The algorithm is iteratively executed until a state of convergence is reached. In a definable interval of iterations, a snapshot of the current population is taken and the final iteration is visually presented as result in a multiple business process model viewer. It is possible to jump back to every stored snapshot in between in order to be displayed in the mentioned user interface. One of the most important key factors for the relevance of the result is the verification of the implementation to assure that it is working in a correct manner without hidden bugs. Therefore, certain software features for the purpose of analysis are required. So, the essential algorithmic steps are made traceable and transparent in order to enable intensive and profound method testing. First, the evolution from generation to generation is verifiable, because the application of genetic operations on the selected individuals is highlighted by color coding the node where a crossover or a particular mutation takes place during every iteration. For example, the correctness of crossover can be checked, while comparing the resulting graphs with the initial ones, shown on demand in the multiple business process model viewer functionality for each generation. Secondly, the underlying mapping that is used as foundation for graph-edit distance calculation between two individuals is stored in the background. In the integrated mapping viewer (see figure 2), those mappings are made visible. Mappings are identified by different colors. The white color is reserved for unmapped nodes. Matching nodes are additionally listed in a table in the right area of the window. An accurate mapping is the main and most important requirement for the correct calculation of the GED. Because of formulating the optimization problem based on the GED as fitness function, the correct and exact calculation has to be verified, apart from the correct implementation of the genetic operations.

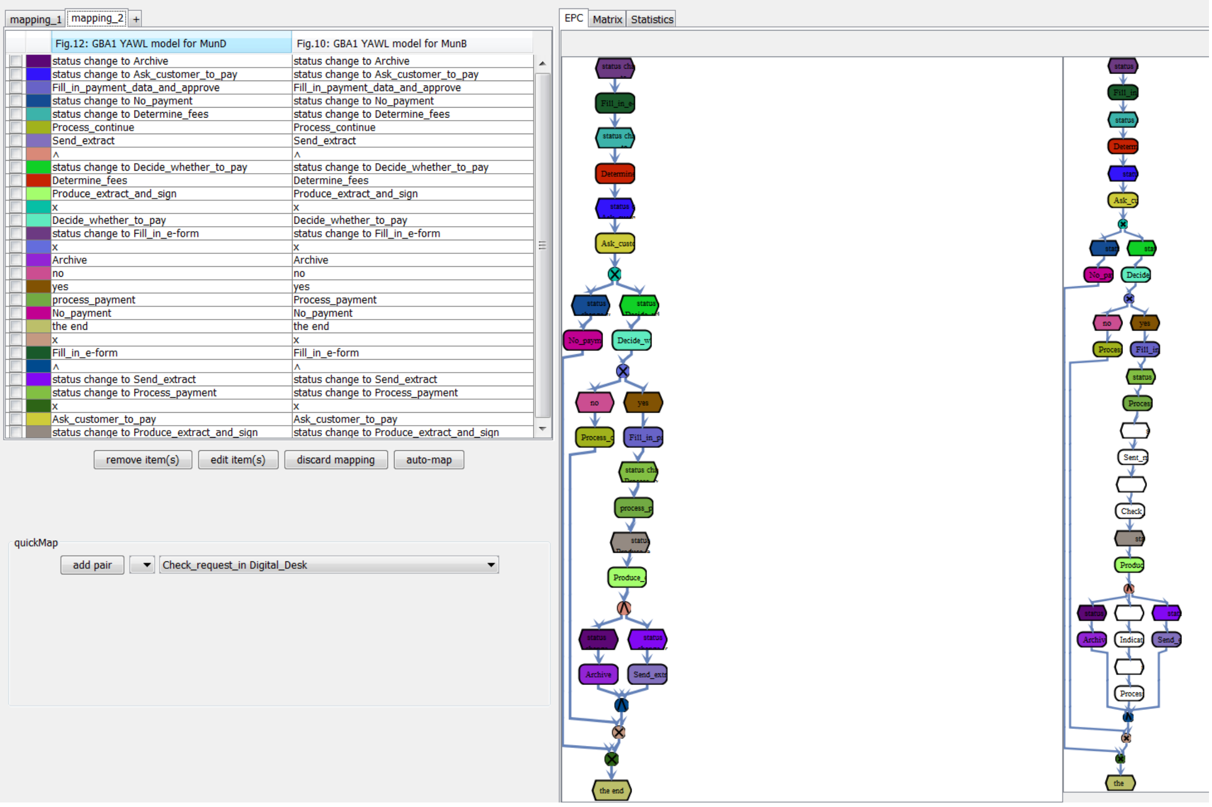


	Fig.12: GBA1 YAWL model for MunD	Fig.10: GBA1 YAWL model for MunB
□	status change to Archive	status change to Archive
□	status change to Ask_customer_to_pay	status change to Ask_customer_to_pay
□	Fill_in_payment_data_and_approve	Fill_in_payment_data_and_approve
□	status change to No_payment	status change to No_payment
□	status change to Determine_fees	status change to Determine_fees
□	Process_continue	Process_continue
□	Send_extract	Send_extract
□	^	^
□	status change to Decide_whether_to_pay	status change to Decide_whether_to_pay
□	Determine_fees	Determine_fees
□	Produce_extract_and_sign	Produce_extract_and_sign
□	x	x
□	Decide_whether_to_pay	Decide_whether_to_pay
□	status change to Fill_in_e-form	status change to Fill_in_e-form
□	x	x
□	Archive	Archive
□	no	no
□	yes	yes
□	process_payment	Process_payment
□	No_payment	No_payment
□	the end	the end
□	x	x
□	Fill_in_e-form	Fill_in_e-form
□	^	^
□	status change to Send_extract	status change to Send_extract
□	status change to Process_payment	status change to Process_payment
□	x	x
□	Ask_customer_to_pay	Ask_customer_to_pay
□	status change to Produce_extract_and_sign	status change to Produce_extract_and_sign

Figure 2: Viewer for the underlying mapping of graph-edit distance calculation

5.2 Scenarios

The evaluation of the described approach is following two different strategies, based on the same real-world business process data set. Within the CoSeLoG project, Vogelaar et al. [27] retrieved 80 business process models (8 different business processes for 10 Dutch municipalities). The GBA1 business process is used as input for the different evaluation scenarios.

- Scenario 1: In a first step, the first variant R* describing the GBA1 business process of the Dutch municipality A is used to generate 20 variants by inserting, deleting and exchanging model elements by a certain probability in order to obtain a diverse synthetic data set of business process models. In a second step, the resulting models are provided as input to the algorithm, deriving inductively a reference model. It is expected that the generated reference model R has to be similar to the GBA1 variant R*, served as pattern for the automatic variant generation. In this scenario, the average number of nodes is 26 and the average number of edges 29. In average, the execution time is below two hours and the number of iterations between 5,000 and 10,000.

- Scenario 2: The 10 individual variants of the GBA1 business process are integrated into one reference model. The generated reference model R is evaluated against a manually developed reference model R* by hand of two independent designers that were separated locally from each other while modeling without knowing the result of the described algorithm and discussing about the modeling task upfront. Afterwards, the different modeled versions have been reasonably merged into a single model based on discussions. While designing the reference model by hand, different practices are possible for creating a reference model. On one side, only the matching nodes and edges can be considered in order to build up a reference model. On the other side, the incorporation of all nodes into some kind of super-model is also a possibility. The compromise on this choice is the selection of nodes that occur with a given probability. The designers followed this way, but in addition, nodes that seemed to be of high importance for the overall process have also been integrated even with a low occurrence. Finally, the result was reviewed and classified as plausible representative. In this scenario, the average number of nodes is 33 and the average number of edges 31. In average, the execution time is below one hour and the number of iterations up to 5,000.

5.3 Measures

For a successful evaluation, meaningful and appropriate measures have to be defined. The most obvious measure is the calculation of the GED between the generated reference model and the expected result, in terms of the distance to the Pareto front. A high GED similarity value within the interval [0, 1] indicates a good result. In the research field of information retrieval, precision and recall are the standard measures for the understanding of relevance. In this paper, the main question is how relevant the obtained reference model is compared to the expected one. This is the reason why it makes sense to transfer these measures as noted in equation 6 and equation 7 from information retrieval area towards the use case of retrieving reference models.

$$\text{precision} = \frac{|R^*| \cap |R|}{|R|} \tag{6}$$

$$\text{recall} = \frac{|R^*| \cap |R|}{|R^*|} \tag{7}$$

In equation 8 the F-measure is defined and interpreted as weighted average (harmonic mean) of precision and recall, rating the accuracy of the retrieved reference model. The values of these measures lie in the interval [0, 1]. A higher value indicates a more relevant and accurate result.

$$F = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{8}$$

5.4 Results

In general, it is difficult to provide a formal proof on the whole purpose of a method. That is the reason why in this paper the presented approach is being evaluated against the following three points:

- Requirements: The stated requirements in section 4 - correctness and efficiency - are fulfilled. The result shows that the reference model is following the event-driven process chain specification. The runtime of the algorithm in order to generate a reference model is in a reasonable relation to the number of individual models. The generation is possible in real-time. This makes the method relevant for practical usage.
- Example of use: The algorithm is applied three times to a different number of different variants. The different relevant evaluation measures are calculated and consolidated as average values in table 1, supported by the visual evaluation of the results that compares the designed to the generated reference model. The numbers do not indicate a significant difference in the comparison of both scenarios. The evaluation was performed on today’s standard hardware (Intel Core i7 vPro 2,3 GHz processor, 16 GB RAM) under Windows 7 64-bit version and Java Runtime Environment 7 64-bit version.
- Proof-of-concept: A software prototype implements our method to simulate, iteration-based, the ongoing development of the initial towards the final population in detail. The single steps are traceable and transparent.

		Event	Function	Operator	All
Scenario 1	GED	0,73	0,70	0,40	0,65
	Precision	0,89	0,88	0,50	0,81
	Recall	0,62	0,58	0,33	0,55
	F-measure	0,73	0,70	0,40	0,65
Scenario 2	GED	0,78	0,86	0,40	0,74
	Precision	0,82	0,90	0,50	0,80
	Recall	0,75	0,82	0,33	0,69
	F-measure	0,78	0,86	0,40	0,74

Table 1: Comparison of the described scenarios based on different evaluation measures

6 Discussion

The proposed idea in this paper operationalizes a reference model as an abstract model that can be transformed in as few steps as possible in as many practices as possible by different edit operations, because it is constructed on the idea of having a minimal graph-edit distance to all individual business process models. It is implicitly assumed by applying the GED that the least common denominator of individual models qualifies as a reference model, which is not necessarily describing a best practice for any organization within the same domain. It may even be the case for a specific organization that only a minority of individual business processes is efficient and effective. More or less, a common practice is developed, which serves as a universally applicable and reusable reference model, helping designers to develop new individual business process models for their organizations based on a representative mindset. The least common denominator as common practice, integrating valid business process variants, referring to the same process, allows this conclusion. In the opposite, a super-model as reference model would provide some form of a superset, providing a large solution space for derived and customized business process models, but cluttering a designer's mindset and preventing him from developing and applying his own creative ideas. Beginning the modeling task given a common practice as template opens up space for extensions while being guided in order to design appropriate business process models for specific organizations in an efficient and effective way. Apart from that, the availability of reference models, based on individual sets of validated business process models, will increase the quality of newly designed processes. However, the availability of high-quality individual business process models [23] or alternative process execution logs [17] as solid foundation is necessary for realizing those advantages.

Discussing the minimal GED calculation leads us to the strongly connected underlying matching problem. An optimal mapping is nothing but an equivalence relation, defined on the set of nodes and edges, minimizing the GED between two graphs. A priori, it has to be satisfied that the individual business processes fit together regarding application domain, forming a representative selection for a common practice. Different similarity measures for establishing such an equivalence relation exists. Behavioral similarity is induced by this representative business process selection upfront, and it is assumed as given that the process flows fit together. Semantic similarity can be considered by embedding linguistic background knowledge a priori. For syntactical similarity, different possibilities exist for quantifying similarity among different labeled nodes. But, regardless how the mapping is determined at the end, the validity and meaningfulness with respect to the application of the GED do not depend on the underlying matching problem.

7 Conclusion

In this paper, we have presented an approach for the inductive development of reference models based on individual business processes. We have compared our approach to other relevant inductive development methods and other related concepts. Furthermore, it is implemented in a software prototype that allows the validation of the algorithm as well as the evaluation of the results. The evaluation is based on real-world business process data and real reference standards, but the evaluation of the results by subject matter experts is still pending. In comparison to that, the validity of the evaluation in this paper is quite low. Nevertheless, the results (cf. section 5.4) show that this innovative conceptual idea in the area of inductive reference model development is promising, even though several mentioned requirements and assumptions have to be fulfilled (cf. section 1.2). In our example in section 5, substitution as one of the possible edit operations (cf. section 3) does not occur, because the mapping between nodes is based on identical labels. That is why the term *subv* is equal to zero in our example by definition of the GED similarity measure in section 3, leading to equality of F-measure and GED similarity measure calculation in this special case (see table 1).

8 Future work

This research is based on the design science paradigm with focusing on evaluating the developed artifact. Involving the target group of subject matter experts in the area of business process modeling for example is stated not to be part of this paper, but is one of the most important next steps. This will be beneficial in terms of underlining the relevance of the described problem as well as having a vast influence on the evaluation itself. Another important step is to address the requirements and assumptions (cf. section 1). In the future, the integration of latent semantics into the existing genetic algorithm framework is an interesting idea in order to avoid the necessary of a priori defined one-to-one mappings by embedding linguistic background knowledge, e.g. the usage of dictionaries, to discover similarities. Also, the introduction of a preceding, cluster-driven analysis is taken into consideration in order to satisfy a greater robustness concerning non-representative selections of individual models. Another point is the extension and improvement of genetic operations. Furthermore, the actual fitness function could be combined with other metrics, and the way how to set up the initial population plays an essential role in evolutionary algorithms. Another direction of extending the current approach is to reduce the complexity of the problem by projecting it in a low-dimensional space under the condition that real-world process data is suitable. It is also a common strategy in other research areas.

9 Literature

[1] Ardalani, P; Houy, C; Fettke, P; Loos, P (2013): Towards a minimal cost of change approach for inductive reference process model development. Proceedings of the 21st European Conference on Information Systems (ECIS).

[2] Becker, J; Knackstedt, R (2004): Referenzmodellierung im Data-Warehousing – State-of-the-Art und konfigurative Ansätze für die Fachkonzeption. Wirtschaftsinformatik 1: 39-49.

[3] Becker, J; Meise, V (2011): Strategy and Organizational Frame. In: Becker, J; Kugeler, M; Rosemann, M (eds.), A Guide for the Design of Business Processes, Springer, Berlin: 91-132.

[4] Becker, M; Laue, R (2012): A comparative survey of business process similarity measures. Computers in Industry 63(2): 148-167.

[5] Bunke, H; Münger, A; Jiang, X (1999): Combinatorial search versus genetic algorithms: A case study based on the generalized median graph problem. Pattern Recognition Letters 20(11-13): 1271-1277.

[6] Chang, CK; Christensen, J; Zhang, T (2001): Genetic algorithms for project management. Annuals of Software Engineering 11: 107-139.

[7] De Madeiros, AL (2006): Genetic Process Mining. PhD Thesis, Technical University Eindhoven.

[8] Dijkman, R; Dumas, M; García-Banuelos, L (2009): Graph matching algorithms for business process model similarity search. In: Dayal, U; Eder, J; Koehler, J; Reijers, HA (eds.), Proceedings of the Business Process Management, LCNS 5701, Springer, Berlin: 48-63.

[9] Dijkman, R; Dumas, M; van Dongen, B; Käärik, R; Mendling, J (2011): Similarity of business process models: Metrics and evaluation. Information Systems 36(2): 498-516.

[10] Ferrer, M; Karatzas, D; Valveny, E; Bardaji, I; Bunke, H (2011): A generic framework for median graph computation based on a recursive embedding approach. Computer Vision and Image Understanding 115: 919-928.

[11] Ferrer, M; Valveny, E; Serratosa, F; Riesen, K; Bunke, H (2010): Generalized median graph computation by means of graph embedding in vector spaces. Pattern Recognition 43(4): 1642-1655.

[12] Fettke, P; Loos, P (2007): Perspectives on Reference Modeling. In: Fettke, P; Loos, P (eds.), Reference Modeling Business Systems Analysis, Idea Group Hershey: 1-20.

[13] Gao, X; Xiao, B; Tao, D; Li, X (2009): A survey of graph edit distance. Pattern Analysis & Applications 13(1): 113-129.

[14] Garey, MR; Johnson, DS (1990): Computers and Intractability; A Guide to the Theory of NP-Completeness.

[15] Gen, M; Cheng, R; Lin, L (2008): Network Models and Optimization Multiobjective Genetic Algorithm Approach, Springer, London.

[16] Gerdes, L; Klawonn, F; Kruse, R (2004): Evolutionäre Algorithmen: genetische Algorithmen – Strategien und Optimierungsverfahren – Beispielanwendungen, Vieweg, Wiesbaden.

[17] Gottschalk, F; van der Aalst, WMP; Jansen-Vullers, MH (2008): Mining Reference Process Models and Their Configurations. In: Meersman, R; Tari, Z; Herrero, P (eds.), On the Move to Meaningful Internet Systems – OTM 2008 Workshops, LNCS 5333, Springer, Berlin: 263-272.

[18] Hevner, AR; March, ST; Park, J; Ram, S (2004): Design Science in Information Systems Research. MIS Quarterly 28(1): 75-105.

[19] Hung, RYY (2006): Business process management as competitive advantage: a review and empirical study. Total Quality Management & Business Excellence 17(1): 21-40.

[20] Jiang, X; Münger, A; Bunke, H (2001): On median graphs: properties, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(10): 1144-1151.

[21] Johnson, R; Pearson, D; Pingali, K (1994): The program structure tree: computing control regions in linear time. PLDI ’94 Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation 29(6): 171-185.

[22] La Rosa, M; Dumas, M; Uba, R; Dijkman, R (2010): Merging Business Process Models. In: Meersman, TSD; Herrero, P (eds.), Proceedings of the International Conference on Cooperative Information Systems, LNCS 6426, Springer, Berlin: 96-113.

[23] Li, C; Reichert, M; Wombacher, A (2010): The Minadept Clustering Approach for Discovering Reference Process Models out of Process Variants. International Journal of Cooperative Information Systems 19(3-4): 159-203.

[24] Mukherjee, L; Singh, V; Peng, J; Xu, J; Zeitz, MJ; Berezney, R (2007): Generalized Median Graphs: Theory and Applications. IEEE 11th International Conference on Computer Vision: 1-8.

[25] Rehse, JR; Fettke, P; Loos, P (2013): Eine Untersuchung der Potentiale automatisierter Abstraktionsansätze für Geschäftsprozessmodelle im Hinblick auf die induktive Entwicklung von Referenzprozessmodellen. In: Alt, R; Franczyk, B (eds.), Proceedings of the 11th International Conference on Wirtschaftsinformatik, Internationale Tagung Wirtschaftsinformatik: 1277-1291.

[26] Song, W; Liu, S; Liu, Q (2008): Business Process Mining Based on Simulated Annealing. The 9th International Conference for Young Computer Scientists: 725-730.

[27] Vogelaar, JJCL; Verbeek, HMW; Luka, B; van der Aalst, WMP (2012): Comparing Business Processes to Determine the Feasibility of Configurable Models: A Case Study. In: Daniel, F; Barkaoui, K; Dustdar, S (eds.), Business Process Management Workshops, Lecture Notes in Business Information Processing 100, Springer, Berlin: 50-61.

[28] Vom Brocke, J (2003): Referenzmodellierung – Gestaltung und Verteilung von Konstruktionsprozessen, Logos, Berlin.

[29] Wegener, L (2005): Simulated Annealing Beats Metropolis in Combinatorial Optimization. Proceedings of 32nd International Colloquium on Automata, Languages and Programming, LCNS 3580, Springer, Berlin: 589-601.

[30] Yahya, BN; Bae, H; Bae, J; Kim, D (2012): Generating Valid Reference Business Model using Genetic Algorithm. International Journal of Innovative Computing, Information and Control 8: 1463-1477.

[31] Zeng, Z; Tung, AKH; Wang, J; Feng, J; Zhou, L (2009): Comparing Stars: On Approximating Graph Edit Distance. Journal Proceedings of the VLDB Endowment 2: 25-36.