SEMI-AUTOMATIC INDUCTIVE DERIVATION OF REFERENCE PROCESS MODELS THAT REPRESENT BEST PRACTICES IN PUBLIC ADMINISTRATIONS

Hendrik Scholta
University of Münster - ERCIS, hendrik.scholta@ercis.uni-muenster.de

Follow this and additional works at: http://aisel.aisnet.org/ecis2016_rip

Recommended Citation
http://aisel.aisnet.org/ecis2016_rip/53

This material is brought to you by the ECIS 2016 Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in Research-in-Progress Papers by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
SEM AUTOMATIC INDUCTIVE DERIVATION OF REFERENCE PROCESS MODELS THAT REPRESENT BEST PRACTICES IN PUBLIC ADMINISTRATIONS

Research in Progress

Scholta, Hendrik, University of Muenster - ERCIS, Leonardo-Campus 3, 48149 Muenster, Germany, hendrik.scholta@ercis.uni-muenster.de

Abstract

In the course of business process management, reference models are widely used for process improvement or as starting point for the creation of individual process models. Current scientific literature mainly offers deductive approaches to construct reference models. Although there are some approaches that inductively develop a reference model from a set of individual process models, these approaches focus on the derivation and representation of common practices. However, there is no inductive method to detect best practices and represent them in a reference model. This paper addresses this research gap by applying design science research to develop an approach for the semi-automatic and inductive derivation of reference process models that represent best practices in public administrations. The approach creates a merged model to keep the structure of the source models and detect identical parts in the process models. It identifies best practices using query constructs and ranking criteria to group process model elements and evaluate these groups. The contribution of this paper is a conceptualization of the approach and a demonstration of its functionality with an example. The implementation and evaluation is subject of future work.

Keywords: Process Management, Process Model Merging, E-Government, Benchmarking, Reference Modeling.

1 Introduction

An option to optimize processes is the alignment of existing processes to best practices. Best practices can be represented by reference process models (Rosemann, 2003). Reference models “are generic conceptual models that formalise recommended practices for a certain domain” (Rosemann, 2003, p. 595). Besides serving as tool for process improvement, reference models may be applied for modeling purposes. Instead of creating a specific process model from scratch, a reference model may be used as basis for the modeling process which is adapted to a specific organization afterwards. This may result in decreased modeling times and costs due to the reuse of existing model segments (Becker and Knackstedt, 2003; Rosemann, 2003).

Due to its importance, a comprehensive amount of research has been performed focusing on reference process modeling in general (e.g. Becker et al., 2007; vom Brocke, 2003; Fettke and Loos, 2003; Fettke et al., 2005; Rosemann and van der Aalst, 2007; Thomas and Scheer, 2006) and in the domain of public administrations (e.g. Algermissen et al., 2005; Baacke et al., 2007; Hinkelmann et al., 2005; Karow et al., 2008). For the design of reference models there are essentially deductive and inductive strategies (Becker and Schütte, 2007). Deductive procedures create reference models specializing general theories and concepts. Creating reference models inductively is the act of generalizing individual models by abstracting from unnecessary details. Although inductive approaches provide empirical evidence and many reference models were developed inductively, there is lack of inductive procedures and most of
the strategies for reference model construction have a deductive nature (Ardalani et al., 2013; Walter et al., 2012). Recently, several automatic or semi-automatic inductive approaches have emerged that focus on visualizing common practices by reference models (e.g. Ardalani et al., 2013; Martens et al., 2015). However, the inductive derivation of reference process models that represent best practices has not been investigated so far. Especially, different definitions of “best” depending on the application context are not covered. Therefore, focusing on the specific domain of public administrations and applying a design science research methodology (Peppers et al., 2007), this paper targets the following research aim: Development of an approach, which semi-automatically derives reference process models that represent best practices in public administrations from a set of individual process models. In contrast to private organizations, public administrations tend to collaboration instead of competition when providing a service (Nutt and Backoff, 1993). This facilitates the share of process knowledge such as process models and the inductive reference model construction.

The approach developed in this paper can be used by process managers, for instance in process harmonization scenarios when different public administrations implement a commonly used software system and in the course of process model libraries that aim at the share of process knowledge (Algermissen et al., 2012; Eid-Sabbagh et al., 2011). The approach is named RefPA as composition of “Reference Model” and “Public Administration”. RefPA takes several individual models from different institutions created with the same modeling language and dealing with the same process as input and returns a reference model. The individual models, which serve as foundation for the reference model construction, are referred to as source models in this paper. Process models can be represented as directed graphs with activities as nodes and control flow relations as edges (Agrawal et al., 1998).

Since the aim of this research is the creation of an IT artifact, this paper applies design science research (DSR) (Hevner et al., 2004) in terms of the DSR methodology proposed by Peffers et al. (2007). Therefore, the following steps have been performed and the paper is structured accordingly: Problem identification (section 1), Derivation of objectives (section 2), Development by a conceptualization (section 3) and Demonstration (section 4). Evaluation is subject of future research. Related work is presented in section 5. Finally, a conclusion and an outlook on future work are given in section 6.

2 Requirements

This section presents objectives for the development of RefPA. Based on literature on characteristics of conceptual models in general and reference models in particular, we derived requirements for an approach for the semi-automatic inductive derivation of reference models that represent best practices:

REQ1: Keeping structure of source models. Due to the inductive nature (Becker and Schütte, 2007), a solution develops a reference model by integrating information of the source models. Thus, the source models predefine the structure of the resulting reference model. The resulting reference model contains only nodes and edges that exist in at least one source model. So, no nodes and edges are included that do not exist in any source model.

REQ2: Identification of commonalities. The resulting reference model contains commonalities of the source models (Ardalani et al., 2013; Rehse et al., 2013). Since they are identical in all source models, these parts represent best practices in any case and will be integrated into the reference model. Consequently, the solution discovers segments that all source models have in common and integrates them into the reference model.

REQ3: Grouping of process model elements. Since best practices are not necessarily common practices (Scheer, 1999), the reference model does not only contain segments that all source models have in common (cf. REQ2). Instead, it additionally incorporates segments that not necessarily the majority of source models has in common but that depict best practices. For this purpose, such segments need to be detected. Since model segments are subgraphs with nodes and edges, the solution offers the opportunity to form groups of nodes and edges in each source model.
REQ4: Evaluation of groups. Since the aim is to derive best practices and detecting the “best” implies a ranking (Bogan and English, 1994), the groups mentioned in REQ3 are evaluated and ranked. As the definition of “best” (e.g. fast, cheap) depends on the aim of a reference model construction and therefore varies from case to case (Rosemann et al., 2011; Schütte, 1998), the solution offers different criteria for the evaluation of groups. The calculations of criteria can make use of annotated information of process model activities such as roles and software systems.

3 Conceptualization

3.1 Merged Model

RefPA’s steps are visualized in Figure 1. In Figure 1, three source models are inputted to RefPA (step 0). In order to fulfill REQ1, a merged model is created (step 1) that contains all nodes and edges of all source models and serves as basis for the reference model, i.e. the reference model emerges from the merged model. The reference model evolves from the merged model by selecting and marking suitable elements. The marked elements will finally be transferred to the reference model. Thus, the merged model determines the positions of the elements in the reference model and therefore the reference model keeps the structure of the source models. The merged model can be created manually or with automated support using approaches from literature (e.g. Gottschalk et al., 2008; La Rosa et al., 2013).

Figure 1. Steps of the Proposed Solution

In order to automatically identify common elements in the source models and fulfill REQ2, the merged model and its elements’ references to their source elements are used (step 2). If an element of the merged model occurs in all source models, then it has a corresponding element in each source model, i.e. the amount of elements it refers to is equal to the amount of inputted source models. In Figure 1, the number of referred elements is exemplarily annotated next to each node. Since node A refers to three source elements and there are three source models, A is a common element. On the contrary, C has two source elements and is not a common element (2 < 3). In order to incorporate all common process segments into the reference model, common elements are automatically marked and will therefore be transferred to the reference model finally in step 5. In Figure 1, marked nodes and edges are highlighted by bold text and lines.

3.2 Query Constructs and Ranking Criteria

In order to group model elements (REQ3) and evaluate these groups (REQ4) in the steps 3 and 4, the user can specify queries which are automatically executed on the source models. The result of each query is a tabular overview with the rankings of the groups. The groups ranked at the top represent best practices (in Figure 1 groups II.1 and III.2) and can be marked by the user to be transferred to the reference model.
RefPA provides query constructs based on SQL to deliver grouping mechanisms (REQ3): GROUP BY, WHERE and CONTAINS. GROUP BY is used to specify the nodes’ properties for the grouping, i.e. which properties such as annotated software systems and executing roles have to be equal for all nodes of a group. For instance, all nodes in a source model that process the same document are grouped. If there are two documents processed in each source model in Figure 1, this results in two groups per model, e.g. L1 for the first document in source model I. All nodes of a group share common values for the properties mentioned in the GROUP BY clause. However, GROUP BY does not ensure certain values for these properties (e.g. the document must be “Application”). Instead, it only enforces that nodes must have equal values for the properties mentioned in the clause.

Whereas GROUP BY is mandatory in a query and is the construct to specify differentiating properties that constitute groups, the other two constructs can be used to define additional conditions that groups have to fulfill. WHERE can be applied to filter the set of groups according to certain properties of their nodes. Hence, certain values and conditions for nodes can be specified which have to be valid for all nodes of a group, e.g. all nodes of a group are not allowed to be processed by a certain role. CONTAINS defines conditions that have to be valid for at least one node of a group. It ensures that a groups contains at least one node with certain property values. For instance, it can be specified that each group has to consist of at least one node that is processed by a certain role. Several CONTAINS statements can be included in a query; each is to be fulfilled by a different node of a group.

In order to fulfill REQ4, RefPA provides (1) a set of ranking criteria and (2) query constructs for evaluation. We derived a set of ranking criteria based on a literature review (Webster and Watson, 2002) and properties provided by activities in process models created with the PICTURE method (Becker et al., 2012). The PICTURE method is a domain-specific modeling language tailored to public administrations and provides a set of 24 standardized and predefined types of activities such as “Formal check”. It allows the enrichment of activities with further properties such as processed documents. The PICTURE method has been widely applied in many German public administrations such as local authorities (Becker et al., 2012; Detemple et al., 2014). The ranking criteria are presented in Table 1.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Documents Source: (Rolón et al., 2006)</td>
<td>Since documents are inputs and outputs of processes in public administrations (Sourouni et al., 2008), this criterion counts the number of documents that are processed by nodes of a group. It can be used to minimize or maximize the quantity of processed documents.</td>
</tr>
<tr>
<td>2 Change of Documents Source: PICTURE document properties</td>
<td>Due to the importance of documents for public administrations (Sourouni et al., 2008), this criterion aims at minimizing the changes of processed documents. First processing document D, then E and finally D again might not be an elegant solution. This criterion is only useful if a group consists of a sequence of connected nodes and does not contain a set of individual nodes widely spread across the model.</td>
</tr>
<tr>
<td>3 Organizational Units Source: (Nissen, 1998)</td>
<td>Analogously to criterion 1, this criterion counts the number of different executing roles of a group’s nodes.</td>
</tr>
<tr>
<td>5 Software Systems Source: (Aversano et al., 2004)</td>
<td>Analogously to criterion 1, this criterion counts the number of different software systems of a group’s nodes.</td>
</tr>
<tr>
<td>7 IT Support Source: (Nissen, 1998)</td>
<td>Since public administrations are encouraged to provide services electronically (European Commission, 2010), this criterion counts how many activities of a group have at least one assigned software system.</td>
</tr>
<tr>
<td>8 External Contacts Source: (Limam Mansar and Reijers, 2007)</td>
<td>In order to minimize the involvement of external participants and to increase thereby the comfort of citizens, this criterion counts how many activities involve an external participant analogously to criterion 7.</td>
</tr>
<tr>
<td>11 Legal Foundations Source: PICTURE Property “Legal Basis”</td>
<td>The services of public administrations are defined and restricted by legal foundations (Nutt and Backoff, 1993). Hence, performing as many legally necessary operations as possible may be a quality criterion. This criterion counts how many activities have at least one assigned legal foundation analogously to criterion 7.</td>
</tr>
<tr>
<td>12 Media Changes Source: (Becker et al., 2010)</td>
<td>A media change is a typical weakness of a process (Delmann and Höhenberger, 2015). Thus, this criterion counts how many activities indicate a media change analogously to criterion 7. For instance, PICTURE offers an according category of node types that can be used to compute a value for this criterion by counting the number of nodes of these types in a group.</td>
</tr>
<tr>
<td>Criterion</td>
<td>Rationale</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>13 Electronic Processing</td>
<td>Since public administrations are encouraged to provide services electronically (European Commission, 2010), this criterion counts how many activities have at least one assigned electronic document analogously to criterion 7.</td>
</tr>
<tr>
<td>14 Paper-based Processing</td>
<td>As public administrations are encouraged to provide services electronically (European Commission, 2010), this criterion counts how many activities have at least one assigned paper-based document analogously to criterion 7 in order to determine a degree of non-digitalization.</td>
</tr>
<tr>
<td>15 Personnel Costs</td>
<td>Since public administrations should use their resources efficiently (Afonso et al., 2010), this criterion calculates the personnel costs of a group’s nodes by considering the nodes’ execution times, the rates of their executing roles and the number of times the nodes are executed.</td>
</tr>
<tr>
<td>16 Lead Time</td>
<td>As public administrations should use their resources efficiently (Afonso et al., 2010), this criterion calculates the lead times of a group’s nodes by considering the nodes’ execution times and the number of times the nodes are executed.</td>
</tr>
<tr>
<td>18 Processing Steps</td>
<td>Since a small number of nodes may indicate a short process, this criterion returns the number of nodes in a group.</td>
</tr>
<tr>
<td>19/ Absolute Frequency MIN/AVG</td>
<td>These two criteria can be used to derive common practices. For each source node of a group, they calculate the frequency of occurrences in the source models. They either return the minimal (“Absolute Frequency MIN”) or average (“Absolute Frequency AVG”) value across frequency values of nodes in a group.</td>
</tr>
<tr>
<td>20 Connectivity</td>
<td>This criterion returns whether the group is a connected subgraph or not.</td>
</tr>
</tbody>
</table>

Table 1. **Overview of Ranking Criteria**

RefPA offers two query constructs to incorporate the ranking criteria: ORDER BY and HAVING. ORDER BY specifies the ranking criteria which are used to rate the groups. The user indicates for each criterion whether groups are ordered in an ascending or descending order. Since the evaluation of groups is essential for RefPA, an ORDER BY clause is mandatory when formulating a RefPA query. HAVING can be used to define conditions for the entire group as own entity that are related to ranking criteria. For example, each group must have five media changes at most or has to be connected when it should be considered for further processing.

Using the above-mentioned constructs, the user specifies a query. RefPA automatically creates a group in each source model for each combination of properties in the GROUP BY clause. For example, grouping according to processed documents and software systems leads to one group per model and combination of document and software. Hence, all nodes that process the same document and use the same software belong to one group. In addition to the nodes, all edges connecting nodes in a group to each other or to nodes, which have already been marked, are part of the group. RefPA only considers groups that fulfill the conditions specified in WHERE, CONTAINS and HAVING clauses. As result of a query, RefPA returns a ranking of the groups for each combination of properties in the GROUP BY clause using the ranking criteria specified in the ORDER BY clause.

Finally, the reference model is automatically assembled (step 5) by removing nodes and edges from the merged model that have not been marked during the detection of identical parts (step 2) or the identification of best practices (steps 3 and 4) such as node C in Figure 1. Process modeling languages such as BPMN (Object Management Group, 2011) use gateways and events to represent control flow relations in more detail. The incorporation of these additional types of nodes that affect the syntactic correctness of a model requires a syntactical revision similar to (Ardalani et al., 2013) in the course of step 5. For example, gateways have to be added if an activity has more than one ingoing or outgoing edge.

---

1 Since they are not relevant for the demonstration in section 4, the following criteria are omitted due to space limitations: Criterion 4 - Change of Organizational Units (Nissen, 1998), 6 - Change of Software Systems (Source: PICTURE software Properties), 9 - External Participants (Source: PICTURE Ext. Participant Properties), 10 - Changes of Ext. Participants (Source: PICTURE Ext. Participant Properties), 17 - Processing Time (Jansen-Vullers et al., 2007).
4 Demonstration

RefPA’s applicability and functionality are demonstrated by a fictional example based on two artificial PICTURE process models visualized in Figure 2. As mentioned above, PICTURE offers 24 standardized types of activities that are graphically illustrated as proposed in (Algermissen et al., 2012; Becker et al., 2012) and their names are given below the graphics written in italic letters. For instance, An1 and Bn1 are of type “Receive document/information”. Based on (Smirnov et al., 2011), we define a PICTURE process model \( m_i \) as tuple \( m_i = (A_i, F_i, P_i, V_i, \text{props}_i) \) with:

- \( A_i \): A finite non-empty set of activities, i.e. nodes
- \( F_i \subseteq A_i \times A_i \): The set of flow relations, i.e. edges
- \( P_i \): A finite non-empty set of properties
- \( V_i \): A finite non-empty set of all occurring property values
- \( \text{props}_i : A_i \times P_i \rightarrow V_i \) is a mapping that assigns a property value to a pair of one activity and one property.

Figure 2 provides the inputted source models (step 0) and their merged model (step 1) that was created manually in this case. In the merged model, the sets of referenced source elements are mentioned within nodes and to the right of edges. For instance, the first node of the merged model refers to the atomic nodes An1 and Bn1. Since there are two source models and the merged node refers to two atomic nodes, RefPA automatically recognizes the first node of the merged model as common element. Figure 2 illustrates the identical parts (step 2) with bold texts and lines in the merged model.

![Figure 2](image-url)

**Figure 2.** Two Exemplary Process Models A and B (Left) and Their Merged Model (Right)
When the user submits a query in the steps 3 and 4, for each source model $m_i$ and combination of properties $j$ in the \texttt{GROUP BY} clause, one group $G_{ij} = (A_{G_{ij}}, F_{G_{ij}})$ with $A_{G_{ij}} \subseteq A_i$ and $F_{G_{ij}} \subseteq F_i$ is created when it satisfies the conditions specified in \texttt{WHERE}, \texttt{CONTAINS} and \texttt{HAVING} clauses. We make use of the calculations for ranking criteria provided in Table 2. $P_{\text{document}}$ ($P_{\text{paperDoc}}$) denotes the set of properties of model $m_i$ relating to (paper-based) documents. During steps 3 and 4, the user is assumed to exemplarily submit the query in Figure 3. The nodes are grouped according to their processed documents. Each group has to contain at least one node with type “Receive document/information” or “Send document/information”. All groups have to comprise three processing steps at most and are ranked ascendingly according to “Lead Time” and “Paper-based Processing”. The automatically generated results are displayed in Figure 3. The group of model B is the best group regarding “Application”, whereas A provides the best group for “Answer”. Hence, the user is assumed to select these two groups to be transferred to the reference model so that the node Bn2 and the edges Be1, Be2 and Ae4 are additionally marked. The resulting reference model is visualized in Figure 4 (step 5). The non-marked elements (An2, Bn5, Ae1, Ae2, Be4, Be5) are automatically removed from the merged model to obtain the reference model.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Criterion} & \textbf{Calculation} \\
\hline
14 & Paper-based Processing \textbf{Algorithmic procedure:} \texttt{score\textsubscript{paper} = 0 } \texttt{For each node} $\in A_{G_{ij}}$: \texttt{if} $|\bigcup_{p\in P_{\text{paperDoc}}}\text{props}(node,p)| \geq 1$ \texttt{OR} $\text{props}(node,Type \ of \ Activity) = \text{Print}$ \texttt{OR} $\text{props}(node,Type \ of \ Activity) = \text{Copy}$ \texttt{then} $\text{score\textsubscript{paper} = score\textsubscript{paper} + 1}$ \texttt{return} $\text{score\textsubscript{paper}}$ \\
16 & Lead Time $\sum_{node \in A_{G_{ij}}} \text{props}(node, \text{Processing Time})$ \\
18 & Processing Steps $|A_{G_{ij}}|$ \\
\hline
\end{tabular}
\caption{Calculations of Applied Ranking Criteria}
\end{table}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{Submitted Query (Left) and its Result (Right)}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.png}
\caption{Resulting Reference Model}
\end{figure}

\section{Related Work}

There are several approaches for the (semi-)automatic inductive reference process model creation. The algorithm by L\textsc{i} \textsc{et al.} (2011) creates a reference process from a collection of process variants and minimizes the distance between the reference model and the variants. The approach allows excluding activities from consideration which have an occurrence frequency in the variants less than a threshold. L\textsc{a} \textsc{rosa} \textsc{et al.} (2013) develop an approach which merges two process models. In addition to creating
a merged model with all activities, it can remove infrequent activities from the merged model and rea-
range the model. Functions based on the minimal graph edit distance are aimed to be minimized in a
heuristic method (Ardalani et al., 2013) and as part of a genetic algorithm (Martens et al., 2014). Since
the graph edit distance is low if many nodes of the different models have a high similarity and can be
mapped, it also depends on the frequency of the nodes. REHSE ET AL. (2013) investigate the usefulness
of abstraction mechanisms for the inductive reference model construction. After abstracting from insig-
nificant elements using one of the mechanisms, a merge algorithm is applied to integrate the remaining
elements. MARTENS ET AL. (2015) present an approach that inductively creates a reference model based
on factor analysis. YAHYA ET AL. (2012) propose an approach to create a reference model out of process
variants using a multi-objective genetic algorithm. By considering cost and duration functions it pro-
vides two further criteria in addition to frequency.

All these approaches create some kind of merged model or develop a reference model based on the
frequency of the activities, i.e. create common practices. However, common practices are not necessarily
best practices (Scheer, 1999). In contrast to the above mentioned approaches, RefPA provides a com-
prehensive set of criteria to cover a wider range of definitions of “best”.

Besides, approaches dealing with process benchmarking using process models (Daneva et al., 1996;
Teuteberg et al., 2013) evaluate entire processes and process models to search for best practices (Cross
and Iqbal, 1995). However, these approaches rate entire models and do not evaluate parts of models.

6 Conclusion and Outlook

The contribution of this paper is to propose an inductive approach to semi-automatically develop refer-
ence process models that represent best practices. It uses a merged model to keep the structure of the
source models (REQ1) and identify commonalities (REQ2) and uses SQL-like query constructs and
ranking criteria to group process model elements (REQ3) and evaluate these groups (REQ4).

This paper is subject to several limitations and potential for further research. As the source models are
from different institutions, a terminological standardization has to be performed before applying RefPA
(e.g. Delfmann et al., 2009). For instance, it has to be ensured that two different terms of property values
do not refer to the same entity such as “invoice” and “bill”. This may be a time-consuming task and it has
to be decided in each case whether it is worth to take the effort and make RefPA applicable or select
another reference model construction method instead. Besides, the control flow and the property values
of the nodes may not always cause the values for the ranking criteria. Instead, other factors such as
different qualities of employees may distort values for criteria such as “Lead Time”. Additionally, it
may not always be meaningful to integrate process segments from different organizations due to differ-
cent circumstances. However, we emphasize that RefPA’s aim is not to create a reference model in a
fully automatic way but to support the user during the construction process. In any case, the user should
reflect the appropriateness of RefPA’s results critically.

In future work, we aim at empirically evaluating RefPA to complete the DSR cycle. Currently, we are
creating a more formal specification of RefPA (e.g. developing calculations for all ranking criteria) and
technically implementing RefPA. Afterwards, we will evaluate RefPA with people from practice and
real-world scenarios in workshops. We aim at acquiring process managers from the PICTURE improve
network (Algermissen et al., 2012). To ensure the comparability of process models from different public
administrations, we will check for synonyms and other linguistic relations between property values in a
preprocessing step. The source models will be collected and standardized by researchers beforehand.
Since other inductive approaches focus on common practices, the practitioners will create reference
models in two ways during the workshops in order to evaluate RefPA: (1) manually and (2) using RefPA.
Moreover, RefPA’s concepts can be transferred to other domains by deriving further ranking criteria in
future research.
References


30th International Conference on Information Systems, Phoenix.


