# Automated Assessment of Process Modeling Exams: Basic Ideas and Prototypical Implementation

Tom Thaler, Constantin Houy, Peter Fettke, Peter Loos

**Abstract:** The assessment of process modeling exercises and exams is a time consuming and complex task. It is desirable to give each student a detailed feedback on their solution in terms of syntactic, semantic, and pragmatic quality. It is obvious that particularly in the case of mass courses with several hundred participants, the individual grading of modeling exams by humans is challenging: Besides reliability, consistency, and validity, the efficiency of the grading process must be guaranteed. Against that background, this paper aims at developing first ideas for an automated assessment of process modeling exams. The objective is to improve modeling education in order to teach students not only to model correctly but to develop *good* models. Our ideas were prototypically implemented and applied in an exemplary scenario with promising results. It was possible to identify important limitations but also to derive reliable semi-automated approaches for the assessment of process modeling exams.

**Keywords:** business process modeling, process model matching, process model understandability, process model evaluation, process model metrics, exam assessment.

## 1 Introduction

Conceptual modeling in general and business process modeling in particular are common tools for designing and managing information systems (IS) in organizations. Furthermore, conceptual models are indeed extensively used in organizational practice [Fe09]. Against this background, many degree programs at German universities dealing with IS in a business context, e. g. *Business Informatics* (BI), put a strong emphasis on *conceptual* and *business process modeling*. Moreover, business process modeling is quite often part of introductory courses to BI which can be attended by students from many different business-related disciplines leading to large business process modeling courses with several hundred participants.

The manual assessment of process modeling exams is a time consuming and complex task. At the same time, it contains some sub tasks, like checking for syntactic issues, which are monotonous with many recurring corrections. An automated assessment could considerably speed up evaluation procedures resulting in a lower expenditure of time needed to assess all exams. Furthermore, it supports - by definition - a consistent and objective evaluation of all modeling exams. It is questionable whether human correctors are able to compete with algorithmic methods in terms of inter- and intra-rater reliability. However, automated assessment can also bear considerable risks. The high degree of modeling freedom of common process modeling languages such as *event-driven process chains* (EPCs) can lead to inconsistent model assessments as there is no *one and only* correct solution for an assignment. In conclusion, there seem to be interesting opportunities and challenges regarding an automated assessment of business process modeling exams. However, so far, no approaches and related experiences in this field are known. Some *related work* has been provided by [Sa12] in the context of automatically assessing "3D modeling" exams or by [NL11] in the context of a general automated assessment of multiple choice exams. However, to our best knowledge, there are no publications concerning an automated assessment of business process modeling exams, which is certainly also one important motivation of the MoHoL 2016 workshop.

Against this background, this paper aims at investigating the potential and challenges of an automated assessment of process modeling exams using a design-oriented research approach [He04]. We present some basic ideas as well as a software prototype for an automated assessment of EPCs. In this context, we first explain *which aspects* of EPC models can be automatically checked and *how* they can be checked differentiating between (1.) *syntactic*, (2.) *semantic* and (3.) *pragmatic* aspects [Li94]. Our prototype will consider all types of aspects and can, furthermore, provide feedback concerning each analyzed process model on this basis. Referring to this feedback, students can not only learn how to design "correct" but also "good" process models which will be easier to understand for human model readers.

This paper is *structured* as follows: after this introduction, we will present some basic ideas for an automated assessment of business process modeling exams in section 2. In this context, we will concentrate on the presentation of particular syntactic, semantic and pragmatic aspects which shall be assessed by our software prototype. In section 3, the prototypical implementation will be introduced and explained in more detail before we present an exemplary application in section 4. Section 5 provides a short discussion of our findings and concludes this article.

## 2 Ideas for an Automated Assessment of Process Modeling Exams

### 2.1 Syntactic Aspects

**Syntactic correctness.** Since the EPC is a non-standardized modeling notation, is it not possible to consistently ascertain the degree of syntactic correctness for arbitrary modeling exams. However, regarding relevant literature, e.g. [Ke92, Me08, Ho14a], there are common criteria for correct EPCs, which can be applied:

(1) Functions have exactly one incoming and exactly one outgoing arc.
(2) Events have a maximum of one incoming and a maximum of one outgoing arc.
(3) An EPC contains at least one start event and at least one end event.
(4) Connectors have at least one incoming arc and at least one outgoing arc.
(5) A connector is either a split connector (one incoming arc, at least two outgoing arcs) or a join connector (at least two incoming arcs, one outgoing arc).
(6) Each function and event label occurs exactly once.
(7) Events and functions do alternate (connectors are skipped).

All these rules can easily be checked automatically without having a sample solution at hand. However, it might be necessary to select the individually relevant rules since there are different perceptions of modeling a correct EPC, e.g. depending on the lecturer.

### 2.2 Semantic Aspects

**Completeness of content.** In the context of process modeling exercises and exams, we assume the availability of a natural language text description of a business process which has to be modeled by the students. Often, one important challenge is to identify all relevant information and contain them in the process model. Since the given textual descriptions are equal for all participants as well as for the lecturer, a “controlled modelling scenario” is given [Th15]. Thus, it can be assumed that the generated process models use a homogeneous terminology for labeling the nodes. Against that background, a promising approach for checking the completeness of the content is the application of recently developed process model matching techniques like presented in [An15]. It is searched for node correspondences between the student solution and the sample solution. The identified correspondences can then be used to quantify the extent, to which a student solution contains the expected nodes in terms of the sample solution.

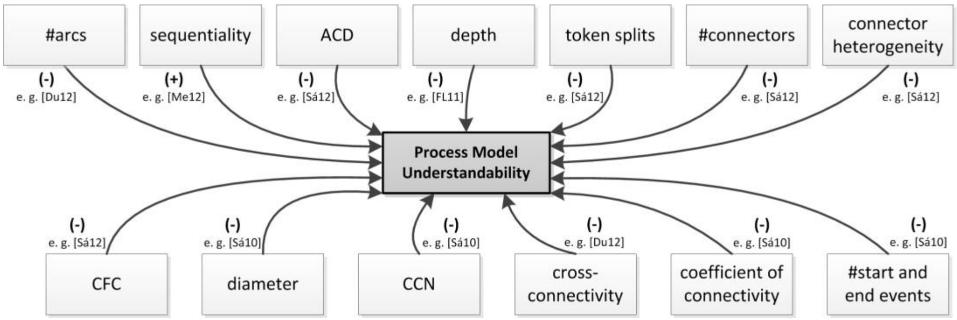
**Semantic correctness.** The semantic correctness addresses the process behavior defined by the process model. In fact, caused by the high degree of modeling freedom, there are different possibilities to model the same process. At the same time, such different solutions may have different state spaces. Thus, indeed, the intended execution traces are covered by the different models but there may also be possible instances which are not intended.

There are different possibilities to measure behavioral correspondence, e.g. by analyzing the state space, the possible execution traces, the behavioral profiles [We09] or the causal footprints [Me07]. Analyzing the state space can be done based on reachability graphs (e.g. [Me08]), which have to be derived from the process model. The reachability graph can then be used to derive the possible execution traces. However, they can also be derived directly, based on the process model itself. The behavioral profile defines three different order relations (strict order relation, exclusiveness relation, interleaving order relation) for each node pair in a process model. In contrast to that, a causal footprint of a process model holds information on the pre-set and the post-set of nodes for all functions in a process model. All mentioned approaches can be used to compare the possible behavior of a process model to the one expected in terms of a sample solution in order to quantify their semantic correctness.

### 2.3 Pragmatic Aspects

**Indicators for attempts to deceive.** One important characteristic of an attempt to deceive in exams or exercises is a high similarity between generated solutions. An easy way to detect possible attempts at deception is to check the similarity between the student solutions using existing process model similarity measures. Hence, since it is assumed that in case of an attempt to deceive a solution is largely copied, a node mapping based on (nearly) identical labels seems to be meaningful. Based on that assumption, a graph edit distance or the percentage of common nodes and edges might be used to determine the similarity between two student solutions.

**Process model understandability.** As a further pragmatic aspect, we aim at an automated assessment of model features related to the pragmatic model quality, especially the understandability of a process model. *Process model understandability* (PMU) is related to the *ease of use* and *effort for reading and correctly interpreting a model*, which is a cognitive process of assigning meaning to the parts of a model [Pa08]. In the last years, there has been a whole host of research trying to identify underlying principles, characteristics or factors influencing the understandability to improve process modeling success. Relevant factors can be related to the model itself, the model reader (personal characteristics) or the modeling language used [Ho14b]. In this research, we concentrate on model-related factors, especially on factors related to the model complexity, which can be automatically assessed. The relationships between computable and well-known complexity metrics and PMU in figure 1 have proven to be reliable in several empirical studies (**(+)** for a positive influence, **(-)** for a negative influence, e.g. “a higher number of arcs has a negative influence on PMU” and “a higher sequentiality has a positive influence on PMU”):



**Legend**: ***#arcs*** / $\boldsymbol{S_A}$=”number of arcs”; ***sequentiality*** / $\boldsymbol{\Xi}$=”degree to which the model is constructed of task sequences”; ***ACD*** / $\boldsymbol{\overline{d_C}}$=”average connector degree describing the average number of input and output arcs of the connectors”; ***depth*** / $\boldsymbol{\Lambda}$=“amount and deepness of nested control structures”; ***token split*** / $\boldsymbol{TS}$=”number of different states after a split connector”; ***#connectors*** / $\boldsymbol{S_C}$=”number of connectors”; ***connector heterogeneity*** / $\boldsymbol{CH}$=”diversity of connector types”; $\boldsymbol{CFC}$ ***/control flow complexity***=“Cardoso’s control flow complexity metric”; ***diameter*** / $\boldsymbol{diam}$=”length of the longest path in a process model”; $\boldsymbol{CCN}$ =”number of nodes connected to a connector”; ***cross-connectivity*** / $\boldsymbol{CC}$=”extent to which all the nodes in a model are connected to each other”; ***coefficient of connectivity*** / $\boldsymbol{CNC}$=”ratio between the total number of arcs and the total number of nodes”; ***#start and end events*** / $\boldsymbol{S_{E_{S+E}}}$=”sum of the number of start and end events”

Figure 1: Some complexity metrics and process model understandability

# 3 Prototypical Implementation

We implemented the above mentioned basic ideas as an integrated model assessment functionality in the RefMod-Miner as a Service.[1] The source is open and publicly available at http://bit.do/RMMaaS. The realization is described in Table 1.

Table 1: Implementation of the syntactic, semantic and pragmatic aspects

| | |
|---|---|
| Syntactic | **Syntactic correctness.** The seven mentioned correctness criteria can be checked automatically. Against the background of the different perception of modeling a correct EPC, it can be selected whether a particular criterion is relevant for the assessment or not. Furthermore, additional criteria can be selected to generate warnings, which are presented in natural language text within in the assessment results. However, the warnings have no influence on the model rating. |
| Semantic | **Completeness of content.** The RefMod-Mine/NHCM algorithm, the overall best performing matching approach of the *Process Model Matching Contest 2015*, is used to compute mappings between the functions of the student solutions and a given sample solution. Based on that, the completeness of content is defined as the recall of activities of the student solution with regard to the sample solution [Th14].<br>**Semantic correctness.** The semantic correctness is operationalized based on the alignment of the possible execution traces [Me08] of the student solutions to the possible execution traces of a sample solution. The true and false positive traces as well as the false negative traces are identified in order to calculate the corresponding precision, recall and f-measure values [Th14]. Traces with matching subsequences are weighted by the fraction of the length of the longest common subsequence. The f-measure value states the semantic correctness. |
| Pragmatic | **Attempts to deceive.** In order to automatically identify attempts to deceive, we applied the percentage of common nodes and edges [Mi07] as a similarity measure. In that context, two nodes are considered as equal if they have the same label. The similarity measure is applied to all student solution pairs. An attempt to deceive is assumed when a predefined similarity threshold of 0.9 is passed.<br>**Process model understandability.** All presented metrics were implemented in order to be able to give students a detailed feedback on the understandability of their generated solution. The metrics are compared to those of a sample solution and consequences for the process model understandability are derived. |

**Overall assessment and parameters.** Relevant aspects (syntactic correctness, completeness of the content, semantic correctness) can be individually weighted for a particular assessment case. Further additional parameters are the max. points and the max. amount of syntax errors which, if reached, sets the points for syntactic correctness to 0. Since the completeness of content and the semantic correctness are values on an interval [0;1], the rating is trivial. Finally, the overall score for a model is the weighted mean of the syntactic and the semantic correctness as well as the completeness of content.

As a result, the tool generates a CSV file containing all partial results as e.g. the number of syntax errors and warnings, the function recall and the semantic precision, recall and f-measure. Moreover, the concrete issues are delivered in the form of natural language texts, e.g. “Function A does not have exactly one incoming edge”. Missing or false positive nodes are explicated by their labels. Furthermore, for each student solution, potential attempts at deception are indicated. In the context of model understandability, a matrix containing all above mentioned metrics for all solutions is generated with an additional “+” for a positive and “-“ for a negative assessment compared to the reference solution.

[1] https://rmm.dfki.de

## 4 Exemplary Application

In order to get a first impression of the performance of the mentioned ideas, we applied the tool to the MoHoL 2016 dataset (http://butler.aifb.kit.edu/MoHoL/). It contains a textual description of a business process, a corresponding EPC reference solution (*ref. sol.*) and 10 EPC student solutions. While using our prototype, we adjusted the weightings and selected relevant modeling rules. The availability of start and end events, the correct event and function syntax and the precise assignment connectors as split or join were selected as mandatory syntax rules, other rules were selected to just generate warnings. Syntactic correctness and completeness of content were weighted with 2, semantic correctness with 1. The maximum number of syntax errors was set to 5 and the overall maximum of points to 100.

The tool was able to correctly detect all syntactic errors respectively warnings. Only 3 of the 10 student solutions offend the selected rules, all of them used incorrect function syntax. With regard to the completeness of content, 7 of the 10 student solutions contained all expected functions. A defect of the implementation can be identified in the case of one of the other three solutions, where functions were not matched because of a

Table 2: Abstract of the assessment result

| **syntactic aspects** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **S01** | **S02** | **S03** | **S04** | **S05** | **S06** | **S07** | **S08** | **S09** | **S10** |
| **#syntax errors** | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 2 |
| **#syntax warnings** | 6 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| **semantic aspects** | | | | | | | | | | |
| **cont. completeness** | 1 | 1 | 1 | 1 | 0,8 | 0,4 | 1 | 0,8 | 1 | 1 |
| **sem. correctness** | 1 | 1 | 1 | 1 | 0,59 | 0,39 | 0,33 | 0,69 | 1 | 1 |
| **pragmatic aspects** | | | | | | | | | | |
| **attempt at deception** | no | no | no | no | no | no | no | no | no | no |
| **model understandability** | | | | | | | | | | |

| metrics | $S_A$ | $\Xi$ | $\overline{d_C}$ | $\Lambda$ | $TS$ | $S_C$ | $CH$ | $CFC$ | $diam$ | $CCN$ | $CC$ | $CNC$ | $S_{E_{S+E}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ref. sol. | 21 | 0,238 | 3,2 | 2 | 5 | 5 | 1 | 5 | 12 | 16 | 0,05 | 1,105 | 2 |
| S01 | 25<br>(-) | 0,72<br>(+) | 3,5<br>(-) | 0<br>(+) | 5 | 2<br>(+) | 1 | 5 | 10<br>(+) | 7<br>(+) | 0,083<br>(-) | 0,962<br>(+) | 0<br>(+) |
| ... | | | | | | | | | | | | | |
| S08 | 20<br>(+) | 0,2<br>(-) | 3,2<br>(+) | 2 | 6<br>(-) | 5 | 2<br>(-) | 4<br>(+) | 12 | 18<br>(-) | 0,088<br>(-) | 1,111<br>(-) | 1<br>(+) |
| ... | | | | | | | | | | | | | |

| | S01 | S02 | S03 | S04 | S05 | S06 | S07 | S08 | S09 | S10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **overall scoring** | **100** | **100** | **84** | **100** | **67,8** | **63,8** | **86,6** | **85,8** | **100** | **84** |

**Legend**: ***#arcs / $S_A$***="number of arcs"; ***sequentiality / $\Xi$***="degree to which the model is constructed of task sequences"; ***ACD / $\overline{d_C}$***="average connector degree describing the average number of input and output arcs of the connectors"; ***depth / $\Lambda$***="amount and deepness of nested control structures"; ***token split / $TS$***="number of different states after a split connector"; ***#connectors/ $S_C$***="number of connectors"; ***connector heterogeneity / $CH$***="diversity of connector types"; ***$CFC$ /control flow complexity***="Cardoso's control flow complexity metric"; ***diameter / $diam$***="length of the longest path in a process model"; ***$CCN$*** ="number of nodes connected to a connector"; ***cross-connectivity / $CC$***="extent to which all the nodes in a model are connected to each other"; ***coefficient of connectivity / $CNC$***="ratio between the total number of arcs and the total number of nodes"; ***#start and end events / $S_{E_{S+E}}$***="sum of the number of start and end events" different labeling style. The student combined two functions, e.g. “Create rejection + inform customer” in one label, which was not assigned correctly. That misinterpretation also leads to a lower semantic correctness. However, the other solutions were scored correctly. One of the most important benefits is the automatic feedback on the models’ understandability. An abstract of the tool output is provided in Table 2, the complete output file is available at http://rmm.dfki.de/docs/MoHoL_2016_Assessment.csv.

## 5 Discussion and Conclusion

Applying our prototype showed that an automated assessment can quickly provide useful information on different correctness and quality aspects and, thus, improve the efficiency of model evaluation processes. However, there are also some *limitations* of an automated assessment: first, the widely acknowledged problems of matching corresponding nodes in process models remains. Although a well-performing matching approach was applied in the assessment tool and although in modeling exams the terminology is provided by textual process descriptions (“controlled modeling scenario”), it was not possible to identify all alignments in the exemplary scenario. That defect leads to problems in automatically judging the semantic correctness and completeness. At the same time, it should be noted that for 9 of the 10 student solutions the matching produced very proper results. Second, the derivation of all possible traces in the current implementation is quite time consuming. In the future, we will investigate the potential of improving the approach’s efficiency also involving causal footprints or behavioral profiles.

In fact, the developed assessment approach was not able to perfectly score all process modeling exams in the application scenario. However, the high potential in some sub tasks could be demonstrated. Thus, we provide a semi-automated assessment approach, which could be used in two variants: (1) all assessment aspects requiring a node mapping are excluded and should be performed manually; (2) node mappings are still automatically generated but have to be verified by a proofreader. However, an automated assessment of process modeling exams always requires the availability of student solutions in a digital and processable format, which is a big hurdle since process modeling exercises and exams are mostly done by hand and on paper and adequate techniques for digitizing paper-based process models are required.

## 6 Literature

[An15] Antunes, G. et al.: The Process Model Matching Contest 2015. In: J. Kolb; H. Leopold; J. Mendling (Hrsg.): Proc. of the 6th International Workshops on EMISA. pp. 127-155.

[Du12] Dumas, M. et al.: Understanding Business Process Models: The Costs and Benefits of Structuredness. In: J. Ralyté et al. (Hrsg.): Advanced Information Systems Engineering (CAiSE 2012), LNCS 7328. Berlin 2012, pp. 31-46.

[Fe09] Fettke, P.: How Conceptual Modeling Is Used. In: Communications of the Association for Information Systems (CAIS) 25 (2009) 43, pp. 571-592.

[FL11] Figl, K.; Laue, R.: Cognitive Complexity in Business Process Modeling. In: H. Mouratidis; C. Rolland (Hrsg.): Advanced Information Systems Engineering (CAiSE 2011), LNCS 6741. Berlin 2011, pp. 452-466.

[He04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: Design Science in Information Systems Research. In: MIS Quarterly 28 (2004) 1, pp. 75-105.

[Ho14a] Houy, C.; Fettke, P.; Loos, P.: Zur Evolution der Ereignisgesteuerten Prozesskette. Multikonferenz Wirtschaftsinformatik 2014 (MKWI). D. Kundisch; L. Suhl; L. Beckmann: 1020-1033. Paderborn 2014.

[Ho14b] Houy, C.; Fettke, P.; Loos, P.: On the Theoretical Foundations of Research into the Understandability of Business Process Models. 22nd European Conference on Information Systems. M. Avital; J. M. Leimeister: 1-26. Tel Aviv, Israel 2014.

[Ke92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". Institut für Wirtschaftsinformatik, Universität Saarbrücken, Arbeitsbericht 89. Saarbrücken 1992.

[Li94] Lindland, O. I.; Sindre, G.; Sølvberg, A.: Understanding Quality in Conceptual Modeling. In: IEEE Software 11 (1994) 2, S. 42-49.

[Me08] Mendling, J.: Metrics for Process Models - Empirical Foundations of Verifiction, Error Prediction, and Guidlines for Correctness. Springer, Berlin 2008.

[Me07] Mendling, J.; Dongen, B. F. v.; Aalst, W. M. P. v. d.: On the Degree of Behavioral Similarity between Business Process Models. Proceedings of the EPK 2007-Workshop. M. Nuettgens; F. J. Rump; A. Gadatsch: 39-58. St. Augustin, Germany 2007.

[Me12] Mendling, J.; Sánchez-González, L.; García, F.; La Rosa, M.: Thresholds for error probability measures of business process models. In: Journal of Systems and Software 85 (2012) 5, pp. 1188-1197.

[Mi07] Minor, M.; Tartakovski, A.; Bergmann, R.: Representation and Structure-Based Similarity Assessment for Agile Workflows. In: R.O. Weber; M.M. Richter (Hrsg.): Case-Based Reasoning Research and Development, LNCS 4626. Berlin, 2007, pp. 224-238.

[NL11] Nettekoven, M.; Ledermüller, K.: Assess the assessment: An automated tool for analyzing multiple choice exams. 4th International Conference of Education, Research and Innovation (ICERI 2011): 2564-2571. Madrid, Spain 2011.

[Pa08] Patig, S.: A practical guide to testing the understandability of notations. Proc. of the 5th Asia-Pacific Conf. on Conceptual Modelling (APCCM '08). Wollongong, Australia 2008

[Sá10] Sánchez-González, L. et al.: Prediction of Business Process Model Quality Based on Structural Metrics. In: J. Parsons et al. (Hrsg.): Conceptual Modeling (ER 2010), LNCS 6412. Berlin 2010, pp. 458-463.

[Sá12] Sánchez-González, L.; García, F.; Ruiz, F.; Mendling, J.: Quality indicators for business process models from a gateway complexity perspective. In: Information and Software Technology 54 (2012) 11, pp. 1159-1174.

[Sa12] Sanna, A.; Lamberti, F.; Paravati, G.; Demartini, C.: Automatic Assessment of 3D Modeling Exams. In: IEEE Transactions on Learning Technologies 5 (2012) 1, pp. 2-10.

[Th15] Thaler, T.; Dadashnia, S.; Sonntag, A.; Fettke, P.; Loos, P.: The IWi Process Model Corpus, Publications of the Institute for Information Systems (IWi), IWi-Heft 199. Institut für Wirtschaftsinformatik. Saarbrücken 2015.

[Th14] Thaler, T.; Hake, P.; Fettke, P.; Loos, P.: Evaluating the Evaluation of Process Matching Techniques. Multikonferenz Wirtschaftsinformatik 2014 (MKWI). D. Kundisch; L. Suhl; L. Beckmann: 1600-1612. Paderborn 2014.

[We09] Weidlich, M.; Mendling, J.; Weske, M.: Computation of Behavioural Profiles of Processes Models. Business Process Technology, Hasso Plattner Institute for IT-Systems Engineering. Potsdam 2009.