Extending and Implementing Process Mining Techniques: Strategies for Detection and Repair Process-Data Quality Issues

1st Kavinda Karunarathna
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e18170@eng.pdn.ac.lk

2nd Jayathu Rajasooriya
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e18276@eng.pdn.ac.lk

3rd Sanjaya De Silva
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e18412@eng.pdn.ac.lk

4th Arthur H.M. ter Hofstede
School of Information Systems
Queensland University of Technology
Brisbane, Australia
a.terhofstede@qut.edu.au

5th Chathura Ekanayake
Queensland University of Technology
Brisbane, Australia
chathura@wso2.com

6th Asitha Bandaranayake
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
asithab@eng.pdn.ac.lk

7th Roshan G. Ragel
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
roshanr@eng.pdn.ac.lk

8th Damayanthi Herath
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
damayanthiherath@eng.pdn.ac.lk

***Abstract*—In contemporary information systems, the execution of processes generates a wealth of historical data stored in diverse formats, including audit trails, system logs, databases, and paper records. Leveraging these data sources through techniques like data mining and process mining enables organizations to extract crucial insights relevant to their respective domains. Process mining, a burgeoning field, utilizes event logs to uncover, monitor, and refine processes. This review paper explores the intricate relationship between process mining and data quality enhancement, with a primary focus on detecting and remedying imperfections in process data. Drawing insights from existing research and practical experiences, the paper elucidates fundamental concepts in event log preparation, including the emergence of imperfection patterns and their role in enhancing the quality of event logs. Through this exploration, the paper provides a comprehensive road map for researchers and practitioners navigating the complexities of enhancing process-data quality, ultimately aiming to empower stakeholders to leverage the transformative potential of process mining within today's data-driven landscape.**

***Index Terms*—process mining, imperfection patterns,event log quality,Event log preparation, Data mining**

I. Introduction

In today's digital era, the execution of processes within information systems generates invaluable historical traces stored in various formats, ranging from audit trails and system logs to databases and paper records. These diverse data repositories serve as rich reservoirs of information, ripe for exploration and analysis. Through the application of advanced techniques such as data mining and process mining, organizations can delve into these data sources to extract vital insights relevant to their specific domains. At the forefront of this exploration is process mining, an emerging field that harnesses event logs to unveil, monitor, and refine processes [24].

The versatility of process mining extends across various domains and organizational contexts, with its impact particularly pronounced in sectors like healthcare. A comprehensive review by Mans et al. [13] underscores the widespread adoption of process mining, showcasing its utility in elucidating crucial aspects of process execution, adherence to guidelines, performance metrics, and opportunities for enhancement. By providing evidence-based insights, process mining offers organizations a comprehensive understanding of their processes, laying the foundation for informed decision-making and strategic improvements.

However, the efficacy of process mining hinges on the quality of input data, as the maxim "garbage in, garbage out" aptly illustrates. Event logs, which serve as the backbone of process mining analyses, must undergo meticulous preparation to mitigate data quality issues. This necessitates a proactive approach to identify and rectify common imperfections within event logs. Herein lies the value of a patterns-based approach, which offers a systematic framework for identifying, classifying, and addressing data imperfections [2].

This review paper aims to delve into the intricate relationship between process mining and data quality enhancement, with a primary focus on detecting and remedying imper-

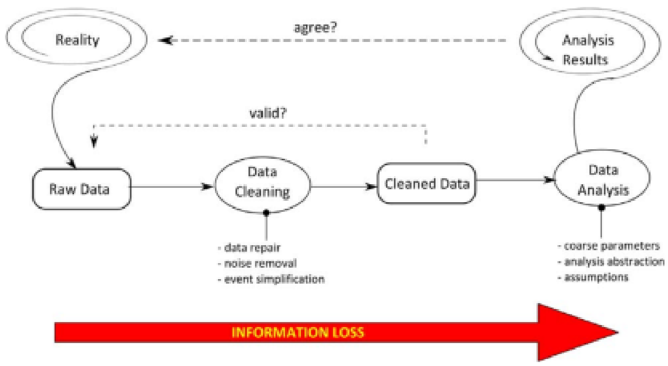

Fig. 1. Garbage-in Garbage-out: the need for data and results validation.

fections in process data. Drawing insights from both existing research and practical experiences, we seek to elucidate fundamental concepts in event log preparation, including the emergence of imperfection patterns and their role in enhancing the quality of event logs. Through this exploration, we endeavor to provide a comprehensive roadmap for researchers and practitioners navigating the complexities of enhancing process-data quality. Ultimately, our objective is to empower stakeholders to leverage the transformative potential of process mining within the dynamic landscape of today's data-driven world.

II. Background

In the process mining field, event logs are essential for analyzing processes, including discovering, conforming, and enhancing them [23]. The success of such analysis hinges on data quality, emphasizing the "garbage-in garbage-out" principle. The paragraph further explores characteristics of event logs, notions of data quality, patterns, pattern languages, and their applications in various domains.

A. Event log basics

CaseID	Attributes			
	Activity	**Timestamp**	**Resource**	**Location**
1	Present at ED	2014-10-03 07:54	Jason	Emergency
1	Triage request	2014-10-03 07:57	Jason	Emergency
1	Triage	2014-10-03 08:03	Susan	Emergency
1	Medical assign	2014-10-03 08:10	Jason	Emergency
1	Blood tests	2014-10-03 09:34	Sarah	Laboratory
1	Admit to hospital	2014-10-03 10:02	George	Ward
2	Present at ED	2014-10-03 08:12	Jason	Emergency
2	Triage request	2014-10-03 08:16	Jason	Emergency
2	Triage	2014-10-03 08:20	Ross	Emergency
2	Medical assign	2014-10-03 08:30	Jason	Emergency
2	Discharge to home	2014-10-03 08:50	Jason	Emergency

Fig. 2. Event log fragment. [20]

An event log consists of cases/traces representing single process executions, with each case containing a sequence of events. Variants refer to unique event sequences within cases. Each case is identified by a case identifier, and each event includes a case identifier and an activity label. Timestamps are often necessary for event ordering. The log may require enrichment with additional attributes for specific analyses, such as resource information for social network analysis. Event logs are typically compiled from multiple raw source logs. The paragraph also defines attributes, events, and event logs, specifying standard attributes like event identifier, case identifier, activity name, timestamp, and resource identifier. Finally, it introduces a formal definition of event logs as a set of events, allowing for relational algebraic operations.

B. Data quality dimensions

To evaluate the quality of an event log for process mining, a suitable quality model is needed. While various data quality models exist in literature, most focus on information systems in general rather than addressing the specific needs of process mining. Data quality is often viewed as a multi-dimensional concept. Wand and Yang summarize the findings of a literature review, categorizing quality dimensions based on their frequency of citation. [26], [27]. Ensuring your data is reliable and trustworthy for decision-making requires understanding its quality. Key dimensions to consider include accuracy, how close data values are to reality (both in format and reflecting the true meaning), and completeness, whether enough data exists for its purpose. Consistency refers to adhering to defined rules within and across different datasets, while currency, timeliness, and volatility reflect how data changes over time. Finally, synchronization ensures proper integration of data with varying timestamps. By evaluating these dimensions, you can identify and address data quality issues, ultimately leading to more informed and accurate decision-making [4].

C. Patterns basics

The concept of using patterns as solutions to recurring problems emerged in the late 1970s thanks to Christopher Alexander's work, particularly "The Timeless Way of Building" [1] and "A Pattern Language: Towns, Buildings, Construction." [2] In these books, Alexander argued that patterns capture common problems and their core solutions, allowing them to be applied repeatedly and adapted to specific contexts. This approach offered a versatile tool for addressing challenges in our built environment. The concept of using patterns as solutions has spread beyond architecture, impacting various fields. The "Gang of Four" [9] popularized this approach in object-oriented programming design patterns. Additionally, experts like Martin Fowler [7], [8] and Grady Booch [3] emphasize that these patterns aren't theoretical inventions but practical solutions discovered through experience. The "Rule of Three" exemplifies this, suggesting a solution becomes a pattern only after successful application in multiple contexts.

III. Related work

This section explores existing research on data quality in process mining. It focuses on both (1) acknowledging the importance of quality event logs and (2) proposing methods to deal with common issues like noise removal. Briefly mentioned is the inadequacy of general data mining approaches for addressing process mining's specific data quality needs.

The Process Mining Manifesto [24] highlights the importance of event log quality, with a star rating system where only high-quality (3-5 star) logs are suitable for analysis. This article proposes ”data imperfection patterns” that capture common issues found in lower-quality logs. By addressing these patterns with their corresponding remedies, the authors aim to empower the community to improve the overall quality of event logs and enable more reliable process mining analyses.

Bose et al [5]. categorize issues affecting process mining event log quality into four main groups: missing data, incorrect data, imprecise data, and irrelevant data. They discuss how these issues can arise in different entities within event logs. While our work aligns with Bose et al.'s categories, we delve deeper into event log imperfections, providing concrete examples based on industry-based logs, particularly in healthcare [19], [21] and insurance [22] domains. Our approach offers a more detailed exploration of event log quality problems, building on our experiences in analyzing such logs.

Mans et al [14]. introduce a process mining data spectrum to categorize event data from different systems within a Hospital Information System (HIS). They highlight the diverse nature of systems within HIS, such as administrative systems and medical devices, which record data with varying levels of granularity and accuracy. Administrative systems often provide coarse-grained timestamps, while medical devices offer fine-grained data, posing challenges for process mining analysis due to either too much or too little detail. Their work describes event log quality along two dimensions: abstraction level and timestamp accuracy, with granularity, directness of registration, and correctness as sub-dimensions of the latter. They use this spectrum to explore data challenges in a case study context and assess if event data from HISs can address medical professionals' queries in process mining analysis. While their approach provides a broad framework, the finer-grained data imperfection patterns proposed in the current article aim to detect and correct specific manifestations of data issues in a more detailed and potentially semi-automatable manner.

When event logs are clean and free of errors, process mining algorithms can build precise models and reveal valuable insights about the underlying process. However, noisy data with irregularities or inconsistencies throws them off course. Instead of accurately reflecting the true process, the resulting models become needlessly complex and misleading. In [11], in describe different types of noise presence in event logs, the author firstly refers to work in [28] on syntactic noise before introducing the notion of semantic noise. Some studies [6], [10], [15] propose use of frequency analysis to detect noise in event logs. They all analyze how often events occur together (directly or eventually) to identify unusual patterns. In particular, [15] uses these frequencies to create rules that capture relationships between events, helping to pinpoint outliers.

Event log imperfection patterns

The extraction of event log imperfection patterns is crucial for preparing raw data logs for process mining analysis. Drawing from diverse fields like software design [9], [29], workflow functionality [25], insider threat protection [16]–[18] and security [12], these patterns offer systematic insights into recurring issues. Each pattern outlines its manifestation, detection strategy, affected data quality dimensions, and consequences on analysis outcomes. Remedial actions are provided along with potential side effects. Indicative rules help identify these patterns, though they don't detail their pervasiveness. For instance, the form-based event capture and collateral events patterns both manifest as multiple events with similar timestamps but in different contexts. These patterns facilitate a structured approach to detect and remediate data quality issues in event logs, enhancing the reliability and usability of the data for process mining analysis. [20]

1) Form-based Event Capture: The Form-based Event Capture pattern refers to a situation where data in an event log is collected through electronic-based forms. Signature of this pattern is, groups of events with the same case identifier and timestamp. Detection involves searching for such groups or looking for 'marker' events with activity names matching form field labels. Regular occurrences of events with the same timestamp indicate the presence of the Form-based Event Capture pattern. Aggregate events with the same timestamp into one event, creating a new attribute to capture relevant data. This reduces parallelism in discovered models. However, the appropriateness of this remedy depends on whether the information collected from the form represents a single process step. [20]

2) Inadvertent Time Travel: The Inadvertent Time Travel pattern captures instances where entries in a log have erroneous timestamps due to the proximity of correct and incorrect data values. This proximity, often occurring when timestamps are very close, can result from human error. Identifying cases with misplaced events, where a strict temporal ordering property is violated. Testing if modifying timestamps based on known rules can correct the event order indicates the presence of this pattern Addressing this pattern requires knowledge of minimum ordering restrictions. Traces violating these restrictions can be identified, and timestamps of events in those traces can be fixed by applying repudiations for standard proximity errors, such as adjusting the timestamp by one day or correcting based on keyboard layout proximity. [20]

3) Unanchored Event: The Unanchored Event pattern occurs when timestamp values in an event log are recorded in a format different from what is expected by the tools used for processing. This discrepancy often arises when constructing the event log from multiple sources, leading to variations in timestamp formats. Signs to detect this pattern include unexpected ordering of activities, unreasonably long or short working/waiting times, missing timestamp information, or values outside the expected range. Remedies for this pattern include preventing tools from misinterpreting timestamp information by adding characters before and after timestamp values to disable built-in interpretation mechanisms. Edit the file using a text editor and apply string manipulation techniques to reformat string values as timestamps. [20]

4) Scattered Event: The Scattered Event pattern involves events in an event log with attributes containing hidden information that can be utilized to derive new events. This hidden information is dispersed across various attribute values of multiple events, providing opportunities to construct additional events. To detect this pattern, manual effort is required for identifying guiding columns (attribute names) and target columns (attributes containing information for new events). 'Marker' values in guiding columns indicate information extraction points, requiring domain expertise for detection. Given the diverse manifestations, a generic solution is unlikely. Once the location of information for new events is known, automated tools can be developed to create these events. [20]

5) Elusive Case: The Elusive Case pattern refers to a log in which events are not explicitly linked to their respective case identifiers. This situation commonly arises in event logs derived from systems that are not process-aware, like GPS tracking systems, web traffic logs etc. The 'Elusive Case' pattern occurs when no attribute exists that can be used as a case identifier, and this can be detected by attempting to discover a process model with randomly tagged attributes as potential case identifiers. Identifying the correct cases to which events belong is crucial. This often involves correlating event log information with data from another source. [20]

6) Scattered case: This imperfection pattern refers to a situation where key process steps are absent in the event log. These missing elements may be recorded in other systems, which gives the ability to construct a complete event log, merged from different sources. This pattern can be detected by abnormal gaps in recorded activities where expected successor activities are missing. This imperfection can be addressed through a record linkage technique, merging events from various sources into one log and properly attributing events to cases. The merging approach depends on the availability of unique identifiers. Where no global unique identifier is determined, false positives and false negatives generated by the linkage algorithm may lead to improper attribution of events to cases. [20]

7) Collateral Events: This pattern describes a situation within an event log where multiple events essentially refer to a specific process step within a case. These duplications may result from the event log being constructed from multiple systems, each with its own way of recording the same activity. It can be detected by groups of activities with very close timestamps, similar or logically consecutive labels occurring within seconds. However, domain experts, who are familiar with system functionality and process steps, are required for a better detection process. The remedy involves developing a knowledge base that specifies which activity names, when occurring together within a short time period, should be merged into a single activity. The knowledge base should include the name of the merged activity and the timestamp to be used. This remedy reduces the total number of unique activities in the event log. [20]

8) Polluted Label: This pattern involves the presence of a group of event attribute values that share a common structure but are distinct due to differences in the specific values that further qualify their meanings. This can be detected by checking the number of distinct values for each attribute and inspecting the values to identify mutable and immutable parts. A semi automated tool can be used to cluster attribute values which can finally lead to reveal this pattern. Imperfection can be repaired by removing mutable words and rearranging the immutable words to form a standard activity name. Mutable words can be preserved as attribute values. [20]

9) Distorted Label: This imperfection occurs when two or more values in the event attribute exhibit strong syntactic and semantic similarities. This generally arises due to incorrect data entry or making unintentional changes to attribute values in data pre-processing step. String similarity search can be applied to detect this imperfection pattern at the basic level. Further complex detection algorithms can be used such as checking for similar, but not identical, consecutive rows in an alphabetically sorted value list. This can be repaired by transforming all the values into uppercase or lowercase. Also, if multiple factors contribute to the existence of the pattern, automated string similarity search can be used to group syntactically similar values and replace them with a single value. Manual intervention may be needed to address semantic dissimilarities. [20]

10) Synonymous Labels: This pattern refers to a scenario where there is a group of values for certain attributes in an event log that are syntactically different but semantically similar. This often occurs when an event log is constructed by merging data from sources with different schemas. Detection of this pattern requires a knowledge base storing 'acceptable' values for each attribute. The pattern exists when two or more attribute values correspond to the same value in the 'acceptable' list. For minor syntactic differences, text similarity search algorithms can be applied to group similar labels and replace them with a predefined value. For substantial syntactic differences an ontology can be used to replace labels with a single value. [20]

11) Homonymous Labels: This pattern characterizes a scenario where an activity is repeated multiple times within a case, with the same activity label applied to each occurrence. However, the interpretation of the activity varies across occurrences due to contextual factors not captured in the event log. Indicators for detection include the presence of numerous arcs and a high ratio of activity occurrences to total cases. Domain knowledge is crucial to conclusively detect the pattern. Remedy for this imperfection pattern can be addressed by explicitly relabeling repeated activities with context-sensitive names. This involves identifying different contexts for activity repetition, developing a formula for context assignment, and differentiating occurrences by adding context information to activity labels. [20]

IV. Research Gap and Further Opportunities

While existing literature presents valuable event log imperfection patterns and remedies for some, a significant gap remains in addressing patterns where solutions are either incomplete or entirely lacking.Identifying gaps in research presents exciting opportunities for innovation in process mining and data quality enhancement. One key gap lies in the development of automated tools to detect and address subtle imperfections in event logs more effectively. While current methods offer insights into common issues, there's a need for advanced algorithms capable of identifying smaller anomalies in large datasets. Additionally, integrating domain-specific knowledge into data quality processes could lead to tailored solutions for different industries. Exploring the potential of machine learning and artificial intelligence in automating data quality tasks is another promising avenue for advancement. Furthermore, longitudinal studies assessing the long-term impact of data quality improvement initiatives on organizational performance are lacking. By addressing these gaps, we can drive innovation and improve decision-making processes across various domains.

V. Conclusion

In conclusion, this review paper has explored the critical intersection between process mining techniques and data quality enhancement, focusing on strategies for detection and repair. Through a comprehensive analysis of existing literature and practical insights, we have identified significant gaps in current research, particularly in addressing nuanced imperfections within event logs. Despite these challenges, we have highlighted exciting opportunities for innovation, including the development of automated tools, integration of domain-specific knowledge, and leveraging machine learning and artificial intelligence. Moreover, the importance of longitudinal studies to assess the long-term impact of data quality improvement initiatives on organizational performance has been underscored.

By bridging these gaps and leveraging these opportunities, organizations can drive innovation and improve decision-making processes across various domains. Enhanced data quality not only ensures the reliability and accuracy of process mining analyses but also facilitates evidence-based insights for strategic improvements. Moving forward, it is imperative for researchers and practitioners to collaborate closely, leveraging interdisciplinary approaches to tackle these challenges effectively. Ultimately, by addressing these gaps and embracing innovative strategies, we can unlock the full potential of process mining in today's data-driven landscape.

References

[1] C. Alexander, The Timeless Way of Building, Oxford University Press, 1979.

[2] C. Alexander, S. Ishikawa, M. Silverstein, A Pattern Language: Towns, Buildings, Constructions, Oxford University Press, 1977.

[3] D. Alur, D. Malks, J. Crupi, Core J2EE Patterns (Core Design Series): Best Practices and Design Strategies, Sun Microsystems Inc, 2003

[4] C. Batini, M. Scannapieco, Data Quality: Concepts, Methodologies and Techniques, Springer, 2006.

[5] J.C. Bose, R.S. Mans, W.M.P. van der Aalst, Wanna improve process mining results? It's high time we consider data quality issues seriously, in: Proceedings of the IEEE CIDM. Singapore, IEEE, 2013, pp. 127–134.

[6] R. Conforti, G. Fortini, M. La Rosa, A.H.M. ter Hofstede, Noise filtering of process execution logs based on outliers detection, Trans. Knowl. Data Eng. PP (99), 2016, 1-1.

[7] M. Fowler, Analysis Patterns: Reusable Object Models, Addison-Wesley, Reading, Massachusetts, 1997.

[8] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.

[9] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Object-Oriented Software, Addison-Wesley, 1994.

[10] L. Ghionna, G. Greco, A. Guzzo, L. Pontieri, Outlier detection techniques for process mining, in ISMIS, volume 4994 of LNCS, Springer, 2008, pp. 150–159.

[11] C.W. Günther, Process Mining in Flexible Environments (Ph.D thesis), Technische Universiteit Eindhoven, 2009.

[12] M. Laverdiere, A. Mourad, A. Hanna, M. Debbabi, Security design patterns: survey and evaluation, in: IEEE Canadian Conference on Electrical and Computer Engineering, 2006, pp. 1605–1608.

[13] R.S. Mans, W.M.P. van der Aalst, R.J.B. Vanwersch, Process mining in healthcare –opportunities beyond the ordinary, Technical Report BPM-13-26, Eindhoven University of Technology, ⟨http://bpmcenter.org/wp-content/uploads/reports/2013/BPM-13-26.pdf⟩, 2013.

[14] R.S. Mans, W.M.P. van der Aalst, R.J.B. Vanwersch, A.J. Moleman, Process mining in healthcare: Data challenges when answering frequently posed questions, in: Process Support and Knowledge Representation in Health Care, volume 7738 of LNCS, Springer, 2012, pp. 140–153.

[15] L. Maruster, A.J.M.M. Weijters, W.M.P. van der Aalst, A. van den Bosch, A rule-based approach for process discovery: dealing with noise and imbalance in process logs, Data Min. Knowl. Discov. 13 (2006) 67–87

[16] A. Moore, M. Collins, D. Mundie, R. Ruefle, D. McIntire, Pattern-based design of insider threat programs, Technical Report CMU/SEI-2014-TN-024, Software Engineering Institute, 2014.

[17] A. Moore, D. McIntire, D. Mundie, The justification of a pattern for detecting intellectual property theft by departing insiders, Technical Report 732, Software Engineering Institute, 2013.

[18] A. Moore, M. Hanley, D. Mundie, A pattern for increased monitoring for intellectual property theft by departing insiders, in: Proceedings of the Conference on Pattern Languages of Programs, 2011, pp. 1–17

[19] A. Partington, M.T. Wynn, S. Suriadi, C. Ouyang, J. Karnon, Process mining for clinical processes: a comparative analysis of four Australian hospitals, ACM Trans. Manag. Inf. Syst. 5 (4) (2015) 19:1–19:18

[20] S. Suriadi, R. Andrews, A.H.M. ter Hofstede, M.T. Wynn, Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs, 2016.

[21] S. Suriadi, R.S. Mans, M.T. Wynn, A. Partington, J. Karnon, Measuring patient flow variations: A cross-organisational process mining approach, in: AP-BPM, LNBIP, Springer, 2014, pp. 43–58.

[22] S. Suriadi, M.T. Wynn, C. Ouyang, A.H.M. ter Hofstede, N. van Dijk, Understanding process behaviours in a large insurance company in Australia: A case study, in: CAiSE, volume 7908 of LNCS, 2013, pp. 449–464.

[23] W.M.P. van der Aalst, Process Mining – Discovery, Conformance and Enhancement of Business Processes, Springer, 2011.

[24] W.M.P. van der Aalst et al. Process mining manifesto, in: Florian Daniel et al. (ed.), BPM 2011 Workshops (1), volume 99 of LNBIP, Springer-Verlag, 2011, pp. 169–194.

[25] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, A.P. Barros, Workflow patterns, Distrib. Parallel Databases 14 (3) (2003) 5–51

[26] Y. Wand, R. Wang, Anchoring data quality dimensions in ontological foundations, Commun. ACM 39 (11) (1996) 86–95.

[27] R. Wang, V.C. Storey, C.P. Firth, A framework for analysis of data quality research, IEEE Trans. Knowl. Data Eng. 7 (4) (1995) 623–640.

[28] A.J.M.M. Weitjers, W.M.P. van der Aalst, Rediscovering workflow models from event-based data using little thumb, Integr. Comput.-Aided Eng. 10 (2) (2003) 151–162.

[29] S. Yacoub, H. Ammar, Pattern-Oriented Analysis and Design: Composing Patterns to Design Software Systems, Addison-Wesley Profession, United States, 2004.