universität wien

BACHELORARBEIT

FUZZY MINER AS AN ADDITIONAL PROCESS MINING ALGORITHM
FOR A
PROCESS MINING VISUALIZATION TOOL IN PYTHON

Author

Albert Krasniqi

01506736

angestrebter akademischer Grad

Bachelor of Science (BSc)

Wien, 2024

Studienkennzahl lt. Studienblatt:	A 033526
Fachrichtung:	Informatik - Wirtschaftsinformatik
Betreuerin / Betreuer:	Dipl.-Ing. Marian Lux

Danksagungen

Ich möchte mich herzlich bei Dipl. -Ing. Marian Lux für seine wertvolle Anleitung und Unterstützung während meiner Bachelorarbeit bedanken!

Meine besonderer Dank gilt auch meiner Familie und meinen Freunden, die mich in jeder Phase meines Studiums unterstützt haben.

Projekt-Repository

https://github.com/albert143/Process-Mining-Visualization

Abstract

Prozess Mining ist eine Technik, um die Prozess-Modelle von Log Files zu extrahieren. Prozesse im reellen Welt sind manchmal sehr unstrukturiert und nicht lesbar, deswegen Prozess Mining wir eine Lösung anbieten, um diese Prozesse in ein lesbare und strukturierte Form darzustellen. Diese Bachelorarbeit stellt die Entwicklung eines Algorithmus vor, welcher Open-Source und Teil einer Desktop-Applikation ist.

Dieser Applikation wurde von Scratch mithilfe von Python als Programmiersprache und verschiedene Librarien aufgebaut. Fuzzy Miner kommt als zweiter Algorithmus in diesem Projekt. Mittels dieser Tool kann man CSV-Files lesen und bestimmte Werte für Events, Cases und Timestamps einsetzen. Das bedeutet, dass die CSV-Files diese drei Werte enthalten müssen, um das gewünschte Ergebnis zu erzielen.

Der Ansatz ist konfigurierbar, deswegen die Benutzer müssen von dieser Applikation erst das Konzept und den Zweck dieser Applikation können, beziehungsweise Fuzzy Miner verstehen, um die gewünschte Ergebnisse zu erhalten.

Konzept der Roadmap ist eine Metapher, die man hier als Konzept verwenden kann. Bei einer Roadmap werden nicht alle Straßen und Objekten dargestellt, sondern nur die, die am wichtigsten sind. Für die Sachen die, nicht wichtig sind, werden bestimmte Abstraktionsregeln angewendet. Das bedeutet, dass die abstrahierende Elemente in einer besonderen Form auf der Roadmap angezeigt werden, oder überhaupt nicht auf der Roadmap erscheinen werden.

Es gibt bereits Tools in denen Fuzzy Miner implementiert ist, aber die sind nicht alle Open-Source (Fluxicon Disco)[1]. Aus diesem Grund wird davon ausgegangen, dass dieses Projekt keine Black-Box sein soll, sondern Open-Source und, dass man diesem Projekt mit anderen Algorithmen erweitern kann. Man kann in diesem Projekt zusätzliche Funktionalitäten hinzufügen oder die Benutzeroberfläche verbessern, da die Benutzeroberfläche nicht die Hauptaufgabe dieser Bachelorarbeit war.

[1] https://fluxicon.com/disco/

Inhalt

1 Motivation

Seitdem Komplexität von Real-Life Prozesse sehr groß ist, ist es auch schwierig diese Prozesse vorstellen, erklären, verbessern und analysieren zu können. Prozess Mining in diesem Fall wird eine Lösung sein, um die Komplexität zu reduzieren und durch Visualisierung einen besseren Überblick zu verschaffen.

Bei großen Projekten existieren zahlreiche Prozesse und komplexe Beziehungen zwischen diesen. Daher ist es zwar grundsätzlich möglich, alle dieser Prozesse zu visualisieren, aber wenn die Logs direkt visualisiert werden, also ohne die zu bearbeiten, wird auch schwierig sein die Prozesse zu lesen und einen Überblick über diesen Prozessen zu haben. Deswegen war es notwendig eine Lösung zu finden, um diese Strukturierung zu vereinfachen, ohne eine falsche Strukturierung vorzustellen, sondern nur die Prozesse basiert auf ein paar Regeln zu abstrahieren.

In dieser Bachelorarbeit wird erklärt, wie man komplexe, wenig-strukturierte Prozesse vereinfachen und richtig visualisieren kann. Die Visualisierung spielt eine große Rolle in diesem Fall. Anstatt Raw Daten bei einer CSV-File zu lesen, werden Graphs erstell. Diese Graphs sehr gut und richtig strukturiert sind. Basiert auf Raw Data von CSV-Files, werden verschiedene Graphs erstellt. Für Fuzzy Miner werden grundsätzlich zwei verschiedene Methoden verwendet. Um diese Methoden zu berechnen, braucht man paar Werten konfigurieren/einsetzen. Danach werden automatisch die Graphs von der Applikation erstellt und in einer leicht verständlichen Form dargestellt.

Es gibt viele Prozess Ming Algorithmen, die schon zu richtigen Ergebnissen führen. Manche existierende Prozess Mining Algorithmen garantieren, dass das resultierende Prozesse „korrekt“ sind. Das Problem ist nicht, dass die resultierende Modelle falsch sind, sondern diese Modelle alle Details zeigen, ohne eine geeignete Abstraktion bereitzustellen. Daher wird das Konzept einer Roadmap als Metapher dafür verwendet, um die resultierende Modelle zu visualisieren.

Einige Projekte wie Disco[2] sind nicht kostenlos zu verwenden. Dies ist auch eine Herausforderung für Menschen, die kein Unternehmen haben und eine solche Anwendung nur für persönliche Gebrauch benötigen.

[2] https://fluxicon.com/disco/

2 Zugehörige Arbeiten

Die Prozess Mining Techniken, die für weniger strukturierte Prozesse geeignet sind, müssen in der Lage sein, eine High-Level Sicht für diese Prozesse zu liefern und alles, was unerwünschten Details zu abstrahieren bzw. zu entfernen. Aktivitäten in einem Prozess können mit Standorten in einer Topologie verknüpft werden (z.B. Städte oder Straßenkreuzungen) und Verbindungen zwischen diesen Städte oder Straßenkreuzungen, in diesem Fall Eisenbahnen oder Autobahnen. [1]

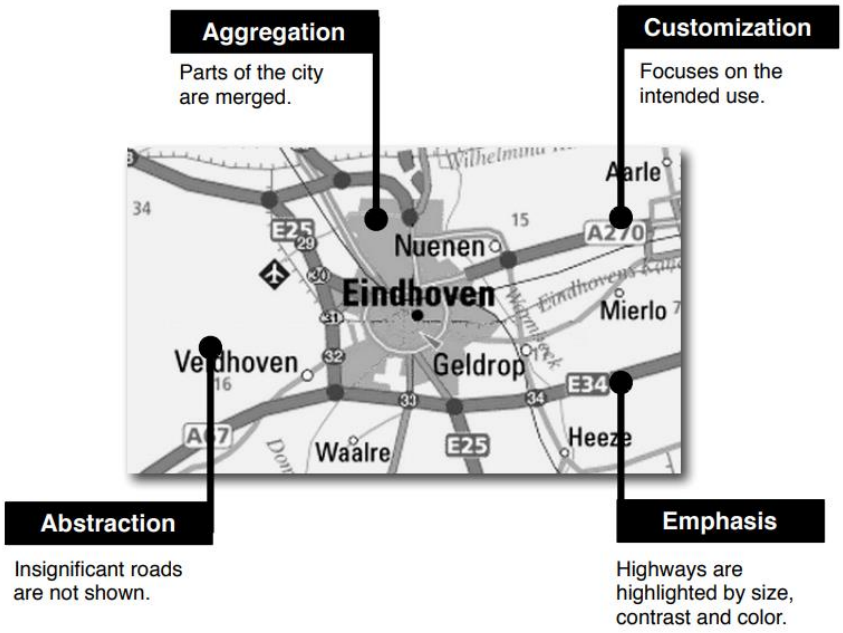


Fig. 1. *Beispiel für einen Fahrplan*

Das Thema Kartographie ist ein Bereich, der sich mit der Erstellung von Karten für einen bestimmten Zweck z.B. eine Fahrradkarte, Fußwegekarte, Straßenbahnkarte usw. Man kann dann, nur einen Teil der Gesamtkarte auswählen und die Dinge, die nicht relevant sind, werden in diesem Fall in der Karte zusammengefasst, abstrahiert oder ganz entfernt.

Betrachtet man thematischen Karten wie das Beispiel in **Fig. 1.** die Lösung, die die Kartographie zur Vereinfachung und Darstellung komplexer Topologien entwickelt hat, so lassen sich daraus eine Reihe wertvoller Konzepte ableiten. In diesem Fall die Topologien zu vereinfachen und darzustellen, wäre eine Möglichkeit wertvolle Konzepte abzuleiten.

Aggregation: Um nicht alle Informationen in der Karte anzuzeigen, die Karte wurde erst in eine aggregierter Form bzw. Cluster umgewandelt. Ein Beispiel sind die Städte in Straßenkarten, in denen bestimmte Häuser und Straßen innerhalb der transitiven Schließung der Stadt zusammengefasst sind. In der thematischen Kartographie wird dieses Konzept zur Verallgemeinerung als Kombination bezeichnet.

Abstraction: Low-Level Informationen werden nicht angezeigt, z.B. sind Fahrradwege für jemanden, der mit dem Auto fährt relevant? Definitiv nicht. Diese Konzept wurde als Abstraktion bezeichnet, d.h. alles, was nicht relevant ist wird aus dem Modell entfernt.

Emphasis: Die Elemente, die wichtiger sind, werden im Modell enthalten.

Customization: Es ist nicht möglich, eine Lösung für alle Fälle zu erstellen. Das bedeutet, dass man zunächst einen Kontext mit bestimmten Detaillierungsgrad und ein bestimmtes Ziel haben muss. [1]

Wenn man dieses Konzept verstanden hat, kann man sich eine Vorstellung davon machen, wie eine mögliche Lösung für einen bestimmten Fall aussehen könnte. Es muss nicht unbedingt ein Fahrplan wie in **Fig. 1.** sein. Es ist lediglich ein Beispiel, das uns helfen kann, das Konzept von Fuzzy Miner zu verstehen. Dafür werden noch Log-Based Prozess Metrics benötigt, um die erwünschte Ergebnisse berechnen zu können.

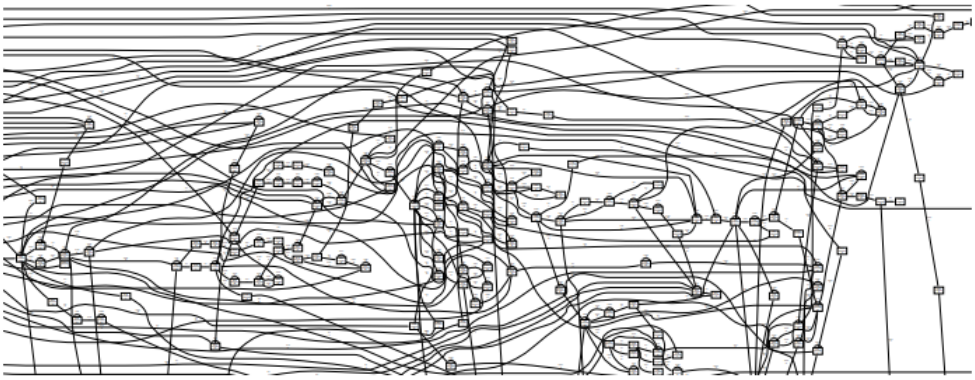


Fig. 2. *Ausschnitt aus einem typischen "Spaghetti"-Prozessmodell (ca. 20% des Gesamtmodells)* [2]

Wenn wir nicht auf dem oben erwähnte Konzept beziehen, sondern wir die Daten von einem Log-Based Prozess direkt, ohne die zu bearbeiten anzeigen, wurde dies als „spaghetti“, bzw. „spaghetti-like“ Prozess Model (**Fig.2.**) bezeichnet, was einen weniger-strukturierten Prozess bedeutet. In der obigen Abbildung sind nur 20% des Modells angezeigt, das bedeutet, dass das gesamte Modell fünfmal größer ist als in der **Fig. 2.** Es ist verständlich, dass es sehr schwierig ist, etwas in diesem Modell zu verstehen. Daher gibt es Algorithmen wie Fuzzy Miner, die es uns ermöglichen, solche Modelle zu vereinfachen und besser lesbar darzustellen. Wobei jedoch zu berücksichtigen ist, dass sich die Richtigkeit des Modells nicht ändert!

2.1 Existierende Tools

Bei existierende Tools kann man auch die Modell mithilfe von Fuzzy Miner vereinfachen. Als nächstes stellen wir zwei Fälle vor, wobei unterschiedliche Methoden zur Vereinfachung der Modelle eingesetzt wurden.

Im Folgenden werden zwei Figuren und Informationen darüber vorgestellt, was für Methoden zur Erstellung dieser Modelle verwendet wurden.

2.1.1 Clustering Methode

Zuerst stellen wir die Clustering-Methode, wobei die Prozesse in drei Arten von Prozessen unterteilt werden. 1.Wichtige Prozesse, 2. Weniger wichtig aber hoch korrelierte Prozesse und 3. Weniger wichtig und wenig korrelierte Prozesse.

Ein ganzes Modell wurde in zwei verschiedene Forms umgewandelt. In der Figur **Fig. 3.** Sieht man zwei verschiedene Ansichten eines Prozesses. Diese Ansichten ändern sich, weil zur Darstellung dieser Ansichten unterschiedliche Log-Based Metrics verwendet wurden.

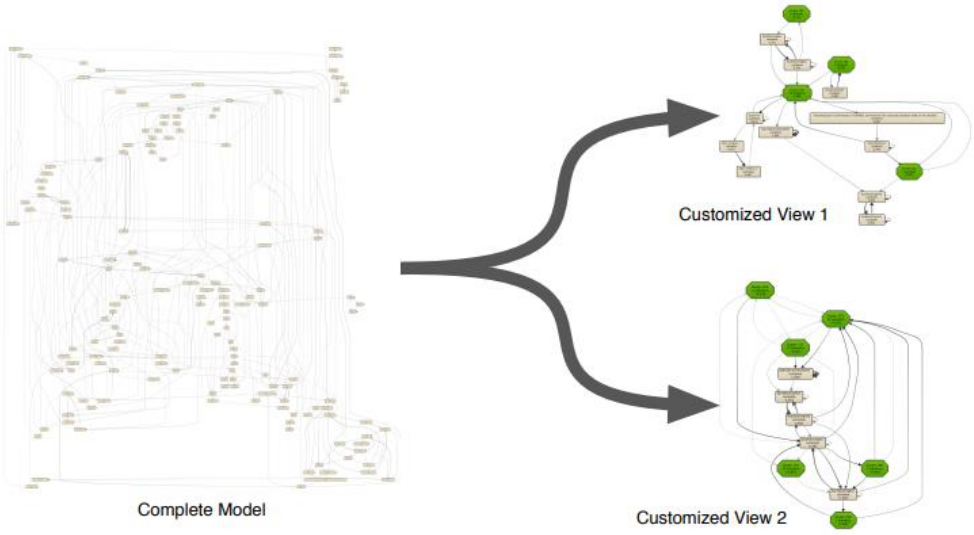


Fig 3. *Vollständiges Modell und zwei angepasste Ansichten eines Prozesses.* [2]

Man sieht auf der linken Seite eine „spaghetti“ Modell, wobei sehr schwer ist die Prozesse zu lesen und auf der rechten Seite sieht man wie sich die Darstellung von den Graphen ändert und ist viel einfacher die Prozesse zu lesen.

2.1.2 Edge Filtering

Um ein Modell mittels Edge Filtering Methode zu filtern, braucht man die Log-Based Metrics, die für Clusterong Methode benutzt wurden, aber auch andere zusätzliche Metrics.

In der folgende Abbildung **Fig. 4.** sieht man eine Beispiel, das mit existierende Tools erstellt wurde. Auf der linken Seite sieht man ein Modell, welches „spaghetti“ Modell bezeichnet wurde und auf der rechten Seite sieht man wie sich das Modell nach der Verwendung von Edge Filtering verändert.

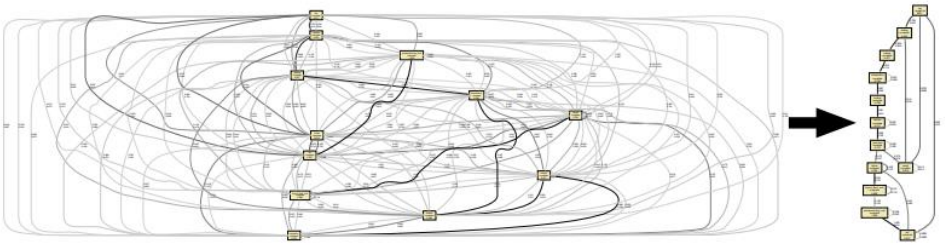


Fig 4. *Beispiel eines Prozesmodells vor (links) und nach (rechts) Edge-Filtering* [2]

Weitere Details zu diesen Algorithmen werden im Abschnitt Algorithmen unten erläutert.

3 Algorithmen

3.1 Fuzzy Miner

In dieser Arbeit werden zwei Methoden erklären, die für die Berechnung von Fuzzy Miner implementiert wurden. Bevor diese Methoden erklärt werden, sollten die Log-Based Metrics erklärt werden. Die sind eine Voraussetzung, um die Fuzzy Miner Algorithmen zu berechnen.

3.1.1 Log-Based Prozess Metrics Unary

Significance und Correlation sind geeignete Konzepts, zur kompakten Beschreibung der Bedeutung des Verhaltens in einem Prozess. Es ist sehr wichtig diese Metrics auf eine angemessene Weise zu messen.

Unser Ansatz ist auf einem konfigurierbaren und erweiterbaren Framework zur Messung von Significance and Correlation basiert.

Es gibt verschiedene Metrics für Significance und Correlation:

- Unary Significance
- Binary Significance
- Correlation
- Binary Correlation

In unserem Projekt wird nur Unary Significance behandelt und Correlation behandelt.

Die unary Significance Metric wird im nächsten Abschnitt eingeführt.

3.1.2 Unary Significance

Unary Significance beschreibt die relative Wichtigkeit einer Event, die als Node im Prozesmodell dargestellt wird. Unser Ansatz basiert auf der Entfernung von Nodes, die weniger Significant sind und aller mit ihm verbundene Egdes von diesem Node.

Eine Metric um die Unary Significance zu berechnen ist Frequency Significance, d.h. je häufiger eine bestimmte Event in einer Log vorkommt, desto mehr Significant ist die.

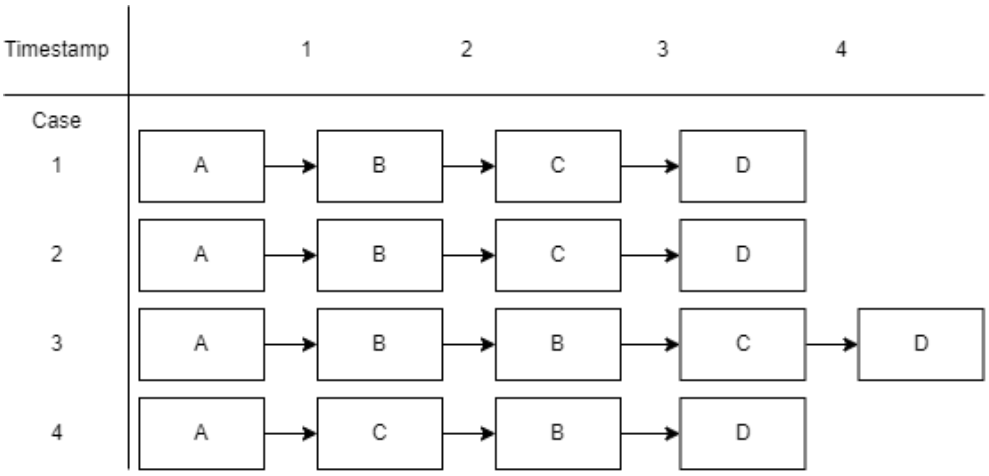
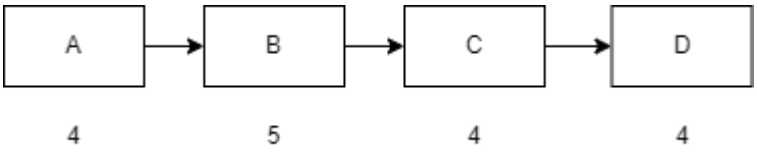


Fig. 5. Ein einfacher Fall mit vier verschiedene Fälle

In der **Fig. 5**. gibt es drei verschiedene Angaben:

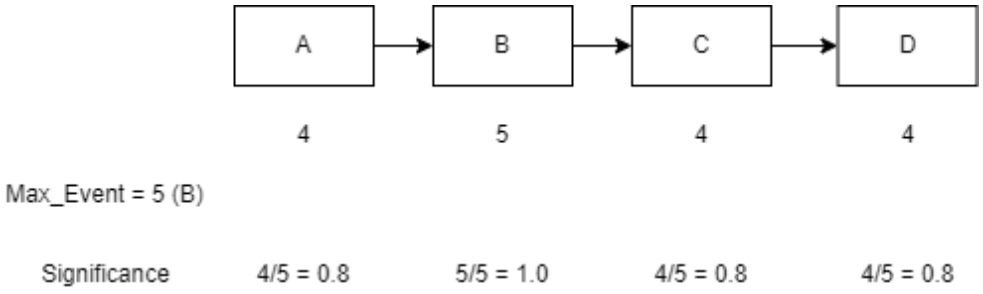
- Timestamp
- Cases
- Events

In diesem Fall Frequency ist:



Erst finden wir das Event, welches die höchste Frequenz hat. In diesem Fall ist das Event B mit Frequenz = 5. Das nennen wir jetzt Max_Event = 5.

Jede Frequenz wird durch Max_Event aufgeteilt:



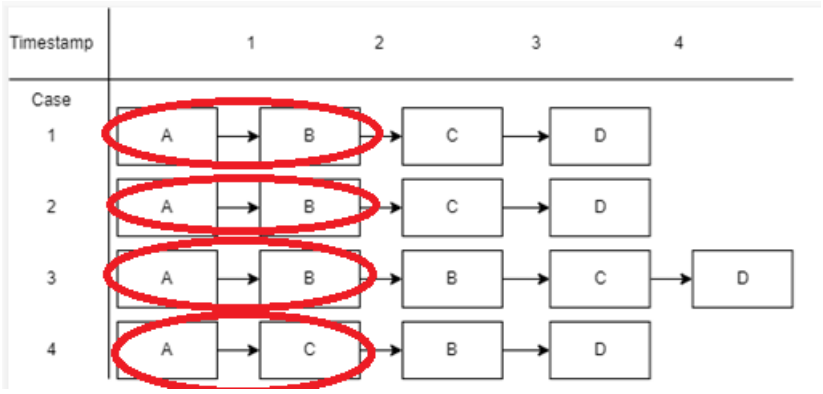
Jetzt haben wir die Significance Metric berechnet, und das wird für die Berechnung von den nächsten Algorithmen als Metric benutzt.

3.1.3 Correlation

Correlation andererseits bezeichnet/misst, wie eng zwei aufeinander folgende Events miteinander sind.

Es wird hier die Berechnung von Correlation von der **Fig. 5.** durchgeführt:

Für alle Evente finden wir ausgehende Edges, in diesem Fall machen wir für das Event A:



A -> X = 4, d.h. das Event A hat vier ausgehende Edges. Das bezeichnen wir als A_outgoing = 4. Jetzt berechnen wir die Correlation zwischen A und mit dem Event A verbundenen Events.

Jetzt berechnen wir die Frequenz für diese Correlierte Events:

A -> B kommt 3-mal vor, d.h. A -> B = 3 und Correlation ist 3/4 = 0.75, wobei 4 ist die Frequenz A -> X in dem Modell.

A -> C kommt 1-mal vor, d.h. A -> C = 1//4 = 0.25.

Gleich berechnen wir die Correlation für alle Events zwischeneinander.

3.2 Clustering Methode

Um die Clustering Methode zu berechnen, brauchen man die Werte für Correlation und Significance haben. In unserer Applikation gibt es ein Slider, wobei beim Starten von der Applikation die Werte werden standardmäßig auf 0.0 eingesetzt. Die kann man später erhöhen und automatisch wird das Modell sich ändern.

Es gibt drei Regeln, um die Clustering Methode zu implementieren:

- 1. Regel: Die hoch signifikante Nodes werden beibehalten.
- 2. Regel: Die weniger signifikante Nodes, die hoch korreliert sind werden zusammengeschart.
- 3. Regel: Die weniger signifikante Nodes, die wenig korreliert sind, werden vom Modell entfernt.

3.2.1 1. Regel

Die hoch signifikante Nodes werden beibehalten.

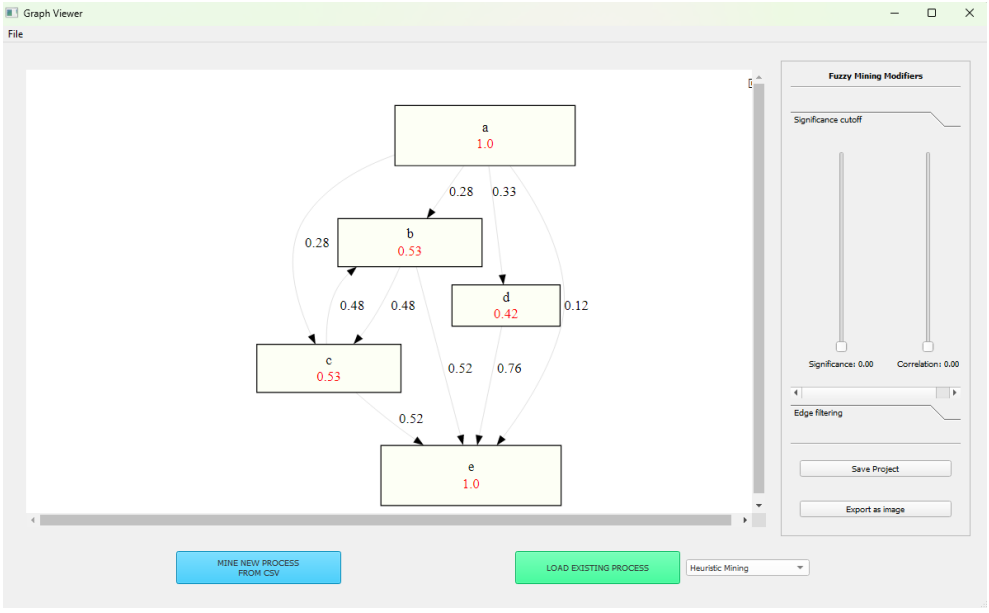


Fig. 6.1. Beispiel beim Starten der Applikation

In der obigen Beispiel sind 141 Fälle in einer CSV-File und so sieht das Prozess nach der Berechnung von Significance und Correlation aus. Auf der rechten Seite sieht man, dass die Werte von Correlation und Significance auf 0.0 gesetzt sind.

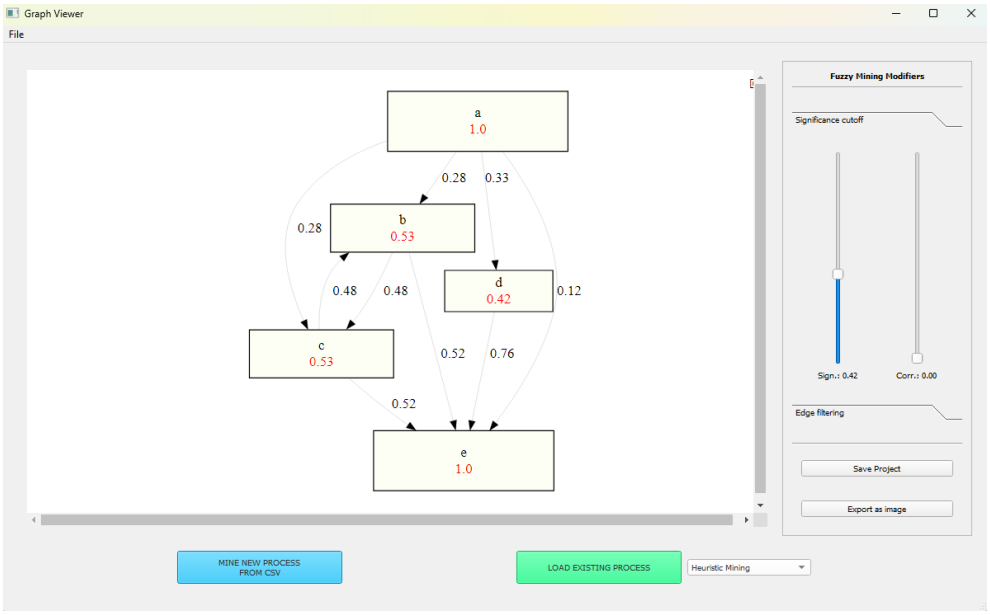


Fig. 6.2. Beispiel, nachdem die Werte für Significance und Correlation geändert wurden

Hier sieht man, wie die Nodes im Prozessmodell beibehalten wurden. Obwohl, die Werte für Significance auf 0.42 und für Correlation auf 0.0 gesetzt wurden, ändert sich das Prozessmodell nicht. Der Grund dafür ist, dass Significance von alle Nodes >= 0.42 ist. Wäre der Wert von Significance 0.43, dann müssten wir die Rule 2 und Rule 3 noch überprüfen.

3.2.2 2. Regel

Die weniger signifikante Nodes, die hoch korreliert sind werden zusammengeschart.

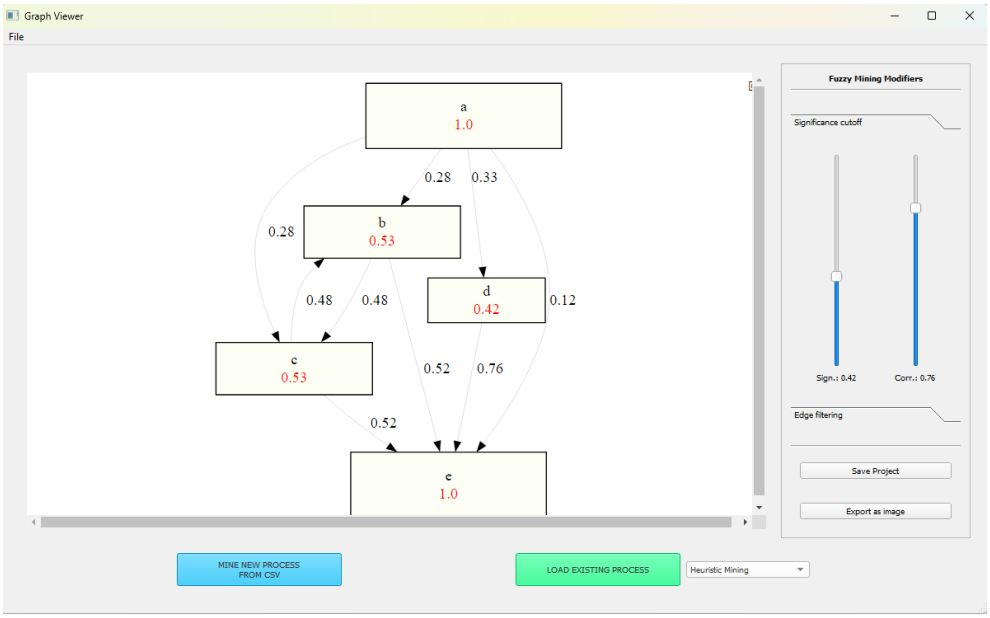


Fig.7.1. Beispiel, Significance >= eingesetzte Significance in Slider

In diesem Beispiel sieht man, dass das Modell sich nicht ändert, weil die Significance von alle Nodes >= als gegeben Significance in Slider auf der rechten Seite ist. Obwohl, Correlation nicht mehr auf 0.0 gesetzt ist, spielt es keine Rolle, weil alle Nodes haben die Significance >= 0.42.

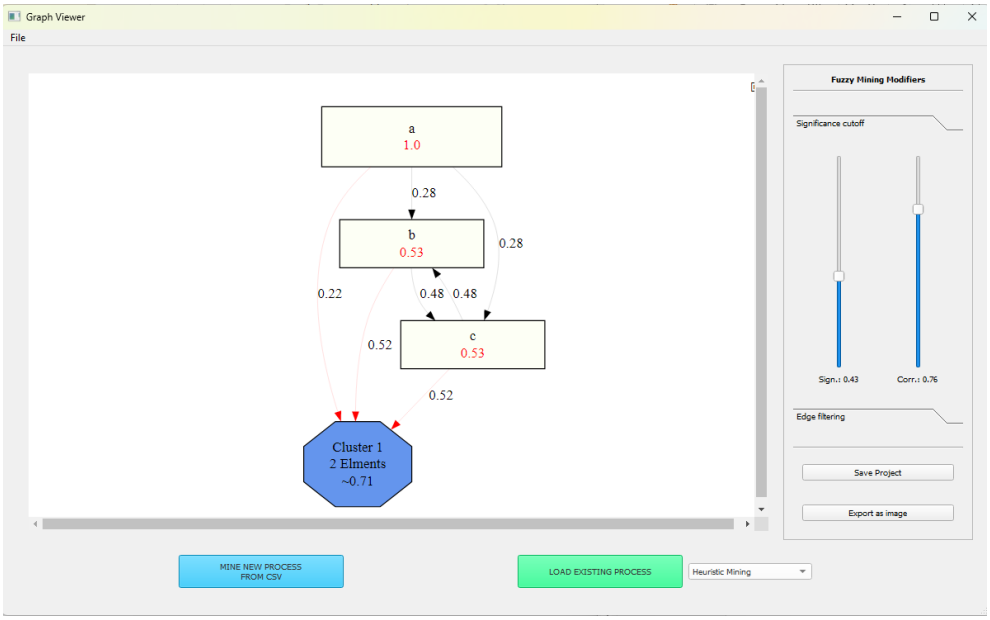


Fig. 7.2. Beispiel, Significance < eingesetzte Significance, aber die Correlaion >= als eingesetzte Correlation in Slider auf der rechten Seite

In diesem Beispiel sieht man wie zwei Nodes zusammengeschart wurden. In der **Fig. 7.1**. ist Correlation zwischen Node d und e 0.76. Deswegen, nach der Prüfung der Regel 2, wurde geprüft, dass diese zwei Nodes die Voraussetzung erfüllen, um in einem Cluster zusammen zu sein.

In diesem Fall werden auch einige Berechnungen durchgeführt, die im Cluster eingefügt werden. Es wird erst die Name mit „Cluster 1“ begonnen und dann die andere Clusters werden inkrementiert. Als zweites Element kommt, die Anzahl der Elemente, die zusammengeschart wurden, in diesem Fall sind 2 Elemente. Und das dritte Element ist die Durchschnittswert der Significance beider Elemente, in diesem Fall 0.42 + 1.0 = 1.42, 1.42/2 = 0.71.

Für eingehende und ausgehende Edges zwischen Nodes und Cluster wird auch die Durchschnittswert berechnet und die Edges werden mit rote Farbe angezeigt.

3.2.3 3. Regel

Die weniger signifikante Nodes, die wenig korreliert sind, werden vom Modell entfernt.

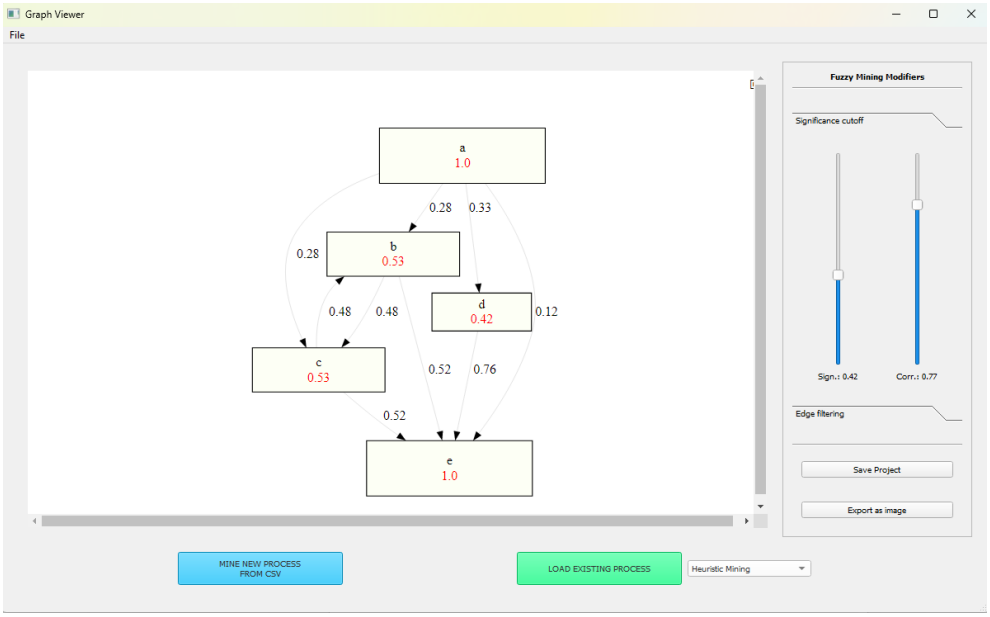


Fig. 8.1. Beispiel, Significance >= eingesetzte Significance und Correlation < eingesetzte Correlation

In diesem Beispiel ist Significane auf 0.42 und Correlation auf 0.77 eingesetzt und das Prozessmodell ändert sich nicht.

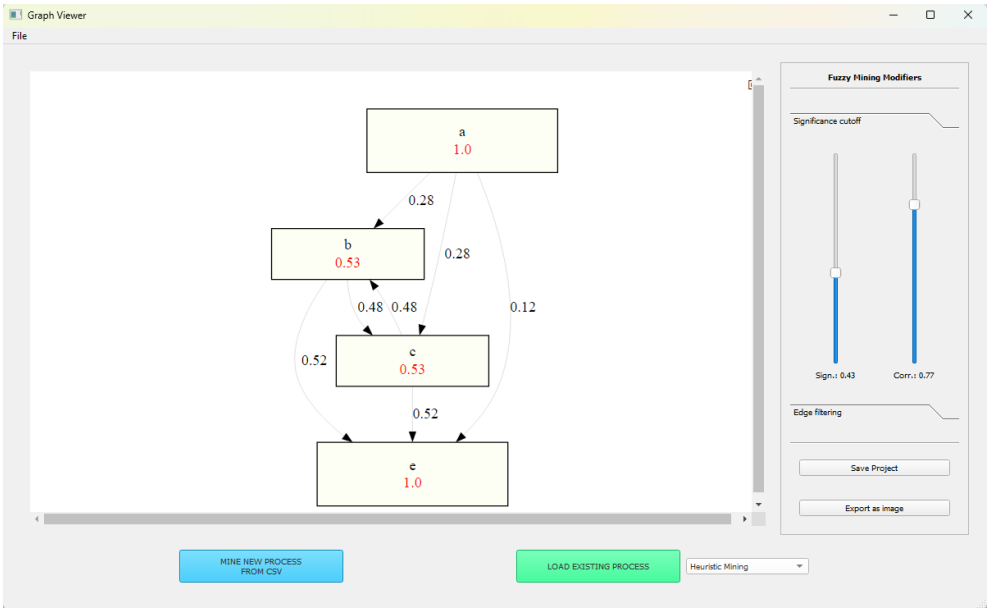


Fig. 8. 2. Significance < eingesetzte Significance und Correlation < eingesetzte Correlation

In vorliegenden Beispiel sieht man, dass die Significance kleiner als eingesetzte Significance ist und auch die Correlation kleiner als eingesetzte Correlation ist, das bedeutet, dass die Nodes, die diese Voraussetzung nicht erfüllen müssen von dem Modell entfernt werden.

In diesem Fall die Node d wird vom Modell entfernt, weil sein Significance und Correlation kleiner als die eingesetzte Significance/Correlation ist. Außerdem, alle eingehende und ausgehende Edges der Node d werden vom Modell auch entfernt.

3.3 Edge Filtering Methode

Um die Methode Edge Filtering zu berechnen, braucht man zusätzlich andere Metrics. Diese Metrics sind auch konfigurierbar, dafür gibt es ein zweites Toolbox auf der rechten Seite, diese Toolbox haben wir Edge Filtering genannt.

Wenn man diese Toolbox öffnen wird, sieht man zwei Sliders. Es gibt einen Slider mit dem Namen „Utility ratio“ und einen weiteren mit dem Namen „Edge cutoff“. Wie gesagt, diese Werte sind gleich wie die Significance und Correlation in den Slider einzusetzen.

Um eine Weitere Struktur aus dem Modell zu ermitteln, müssen wir die verbleibende Edges durch Edge Filtering Methode filtrieren.

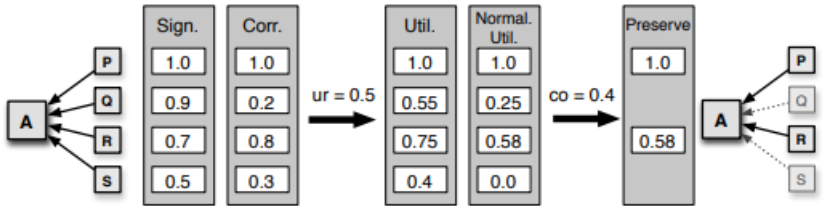


Fig. 9. Filterung der Menge der eingehende Edges für eine Node A [2]

Diese drei Formeln braucht man, um die Util und normalised Util Werte zu berechnen:

- util (A, B) = ur · sig (A, B) + (1 − ur) · cor (A, B)
- Normalised Util: NU = $\frac{U-MinU}{MaxU-MinU}$, U-util value
- Konfigurierbare edge cutoff co ∈ [0, 1] und utility ratio ur ∈ [0, 1]

Daher bewertet unsere Edge Filtering Methode jede Edge A -> B nach ihrem utility (A, B), einer gewichteten Summe aus ihrer Significance und Correlation. Ein konfigurierte utility ratio ur ∈ [0, 1] bestimmt die Gewichtung. [2]

Der zweite Parameter ist Edge Cutoff, welche die Aggresivität des Algorithmus bestimmt, d.h. je höher sein Wert ist, desto mehr entfernt der Algorithmus Nodes. In sehr unstrukturierten Prozessen ist es mehr sinnvoll niedrige Wert für die utility ratio einzusetzen, so dass Correlations stärker berücksichtigt werden und solche mehrdeutigen Situationen auflösen. Darüber hinaus wirkt ein höhere Wert von Edge Cutoff als Verstärker, welcher die wichtigsten Nodes zu unterscheiden hilft. [2]

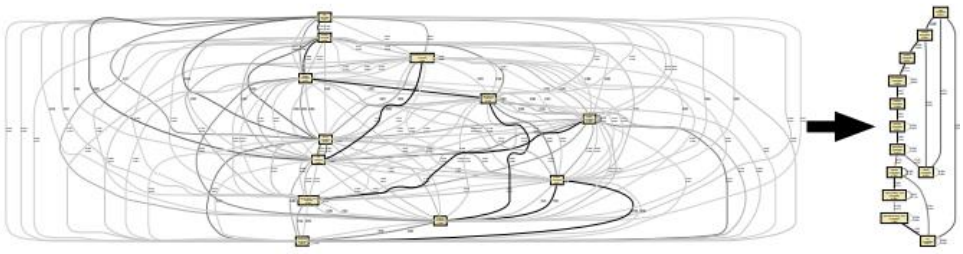


Fig. 10. Beispiel eines Prozessmodelles vor (links) und nach (rechts) Edge Filtering [2]

Fig. 10. Zeigt die Auswirkung der Edge Filtering auf einen kleinen, aber sehr unstrukturierten Prozess. Die Anzahl der Nodes bleibt gleich, während das Entfernen einer geeigneten Teilmenge von Nodes bringt eindeutig Struktur in das zuvor chaotische Prozessmodell. [2]

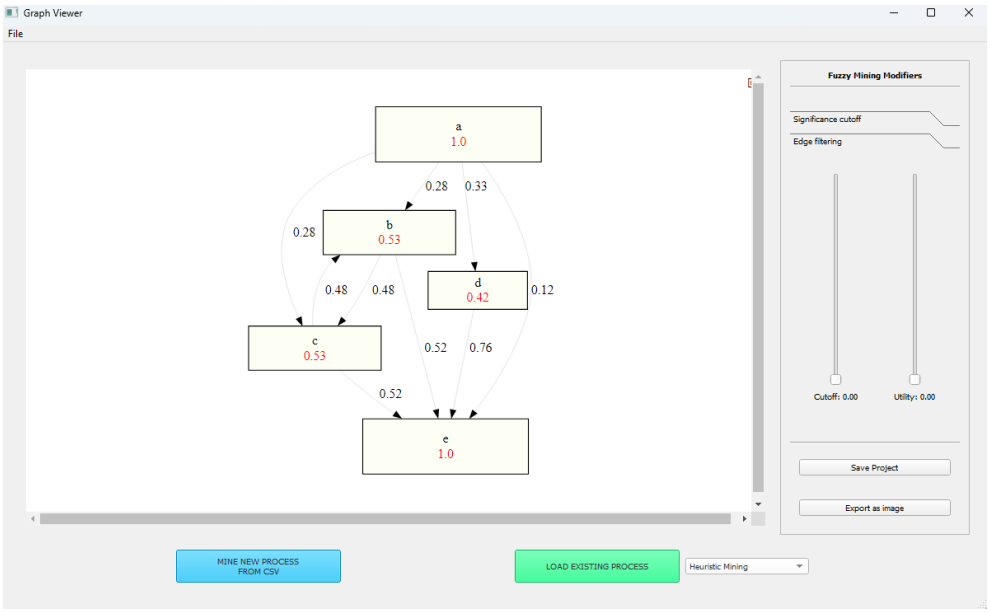


Fig. 11.1. Beispiel, wobei Utility ratio und Edge cutoff auf 0.0 gesetzt sind

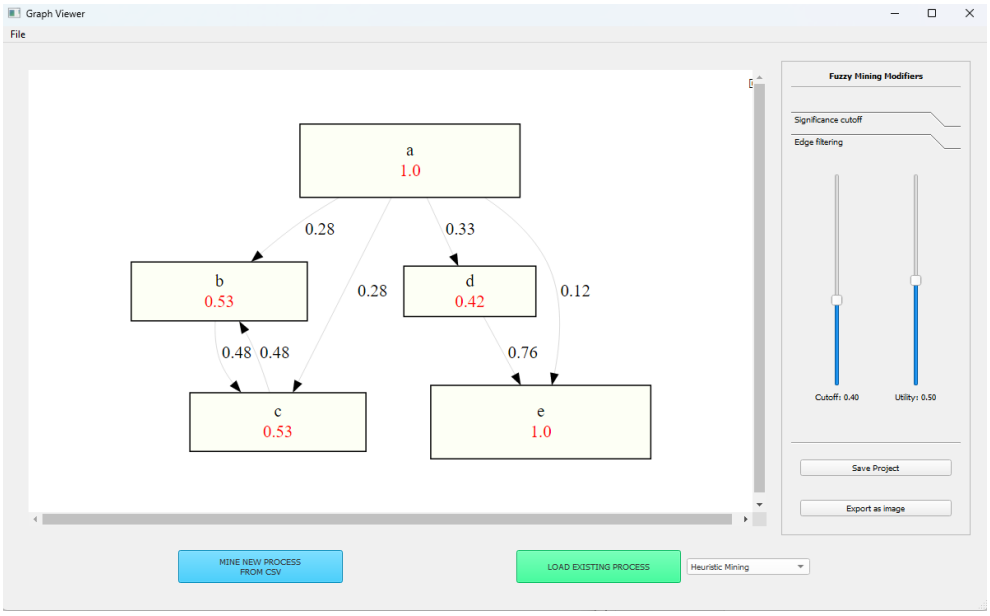


Fig. 11.2. Beispiel, wobei Utility ratio den Wert 0.5 und Edge cutoff den Wert 0.4 hat

In der Fig.11.1. sind insgesamt 9 Edges. Nachdem wir den Wert 0.5 für ur und 0.4 für co eingesetzt haben, in der Fig. 11.2 findet man nur 7 Edges.

In den nächsten Beispiel werden wir den Wert 0.8 für Edge cutoff einsetzen, um zu schauen, ob andere Edges vom Prozessmodell entfernt wurden.

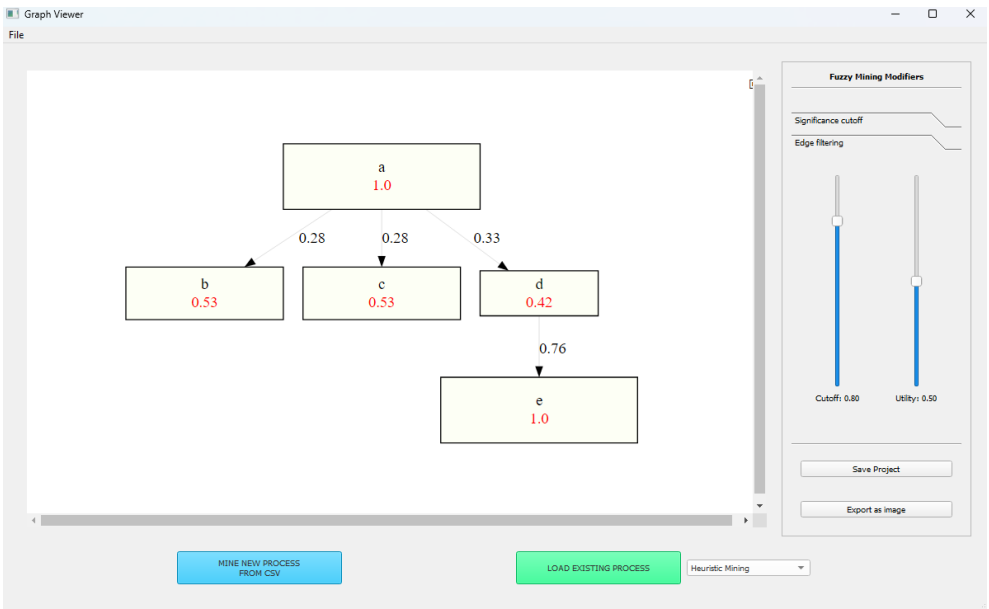


Fig. 11.3. Beispiel, Edge cutoff auf 0.8 eingesetzt, Utility ratio bleibt unverändert

Nach Änderung des Wertes für Edge Cutoff werden im Gegensatz zu Fig. 11.2. drei weitere Edges entfernt.

4 Implementierung

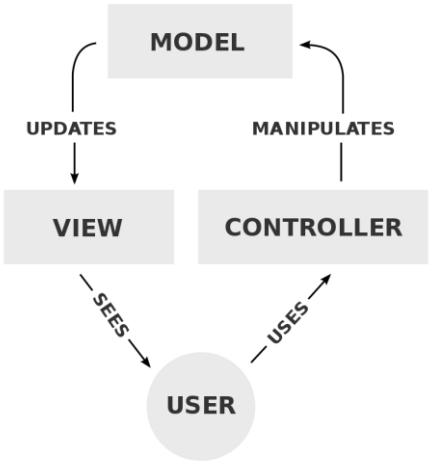


Fig. 12. MVC-Pattern[3]

Bei dieser Applikation gibt es drei Hauptkomponenten:

- Das Model
- Die View
- Der Controller

Diese Konzept nennt man MVC-Architektur. Die Idee dahinter ist, die Trennung von Anwendungslogik, Benutzeroberfläche und die Datenvereinbarung zu ermöglichen.

In unserem Fall das Model ist die Klasse fuzzy_mining.py, die View ist die Klasse fuzzy_graph_view.py und der Controller ist die Klasse fuzzy_graph_controller.py.

[3] https://en.wikipedia.org/wiki/File:MVC-Process.svg

Das Model enthält die Anwendungsdaten. Die View ist nur für die Darstellung der Daten verantwortlich und weiß es nicht die Bedeutung von den Daten oder wie die implementiert sind. Die View einfach ist für die Visualisierung von Graphs verantwortlich. Der Controller verarbeitet die durch Ansicht ausgelöste Events, manipuliert das Model und stellt einen Ansicht mit richtigen Informationen dar.

5 Evaluation und Diskussion

Die Applikation habe ich mit einer Kollegin getestet. Ich habe erst ihr grundlegende Informationen erklärt, danach hat sie die Informationen auf ReadMe gelesen und das Programm gestartet. Sie hat eine CSV-File ausgeführt und jedoch war ihr am Anfang nicht ganz klar, worum es genau geht. Nachdem ich ihr mehr Informationen über Process Mining als Konzept erklärt habe, konnte sie das Konzept in weniger Minuten verstehen. Das Roadmap Konzept war besonders logisch, da es sich um ein echtes Beispiel handelt, das auch Menschen ohne spezifische IT-Kenntnisse vertraut war. Danach war für sie auch klar, warum sich die Prozesse nach der Änderung der Werte in den Slider verändern.

Bei der Implementierung kam auch chatGPT zum Einsatz, da dies mein erstes Programm in Python war. Früher habe ich meistens Programme geschrieben, die mit Java implementiert wurden. Und manchmal im Gegensatz zu Java gab es in Python Syntaxunterschiede, die ich über chatGPT sehr einfach zu suchen waren.

6 Schlussfolgerungen und zukünftige Arbeit

In der Umsetzung dieses Projekts habe ich mein Bestes gegeben, jedoch gibt es Aspekte, die in Zukunft verbessert werden könnten. Insbesondere das Benutzerinterface könnte optimiert werden. Da dies jedoch keine zwingende Anforderung war, lag mein Hauptaugenmerk auf der Funktionalität.

Man könnte auch mehr Interaktion mit Graphen implementieren, momentan kann man nur die Daten von einem Node bzw. Edge lesen. In der Zukunft könnte es eine Voraussetzung zu implementieren sein.

Man könnte auch für Fuzzy Miner andere Metrics wie Binary Correlation und Binary Significance einfügen.

Derzeit hat der User die Möglichkeit eine CSV-File mittels Fuzzy Miner oder Heuristic Miner auszuführen. In der Zukunft wäre es denkbar, weitere Algorithmen bei der Applikation zu integrieren.

7 Bibliographie

[1] C. W. Günther, "(2009), Process mining in flexible environments. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Industrial Engineering and Innovation Sciences]. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR644335".

[2] C. &. A. v. d. ,. M. P. Günther, "(2007) Fuzzy mining - adaptive process simplification based on multi-perspective metrics. Proceedings of the 5th International Conference on Business Process Management (BPM 2007) https://doi.org/10.1007/978-3-540-75183-0_24".

[3] W. M. P. v. d. Aalst, Process Mining, Discovery, Conformance and Enhancement of Business Processes.