



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

KATEDRA INFORMATYKI STOSOWANEJ

Modelowanie Biznesowe

Klasteryzacja, filtrowanie i przetwarzanie logów OCEL

Autor:

Olaf Zdziebko, Wojciech Żyła

Kierunek studiów:

Informatyka i Systemy Inteligentne - Inżynieria Oprogramowania

Opiekun pracy:

dr inż. Krzysztof Kluza

Kraków, 2023

Spis treści

1. Wstęp.....	5
1.1. Cel projektu	6
2. Filtrowanie i wizualizacja logów OCEL	9
2.1. Wczytywanie logów	9
2.2. Filtrowanie i wyświetlanie informacji.....	9
2.2.1. Podstawowe informacje o logach.....	9
2.2.2. Filtracja po id zdarzenia i rodzaju obiektu.....	10
2.2.3. Wyświetlanie zdarzeń powiązanych z wybranym klientem.....	11
2.2.4. Sprawdzenie ilości zamówień w danym dniu	12
2.2.5. Podgląd interakcji wybranego obiektu z innymi obiekta	12
2.2.6. Wyświetlanie obiektów powiązanych z wybranym obiektem	13
2.2.7. Podgląd zakończenia cyklu życia obiektu	13
2.3. Testy przeprowadzone dla innego logu	13
2.4. Wizualizacja logów w postaci sieci Petriego.....	17
3. Klasteryzacja logów OCEL.....	19
3.1. Rozróżnianie logów.....	19
3.2. Algorytm klasteryzacji	19
3.3. Wizualizacja klastrów.....	19
3.3.1. Order Management log	19
3.3.2. Recruiting Process (Simulated).....	20
3.3.3. SAP ERP IDES instance - Transfer Order (Storage).....	22
4. Podsumowanie.....	23

1. Wstęp

Dzienniki zdarzeń są punktem wyjścia do zastosowania technik eksploracji procesów. Klasyczne dzienniki zdarzeń (ang. logs) składają się ze zdarzeń, które charakteryzują się identyfikatorem przypadku, odnoszącym się do unikalnej instancji procesu. Jednak w rzeczywistości mamy do czynienia z systemami informatycznymi, takimi jak systemy SAP ERP, które zawierają wiele pojęć przypadków, np. zamówienia, pozycje i pakiety. Można z nich wyodrębnić obiektowe dzienniki zdarzeń (tj. dzienniki zdarzeń z wieloma pojęciami przypadków).

Standard dla logów obiektoorientowanych OCEL (ang. Object-Centric Event Log) powstał, by zapewnić ogólną standaryzację wymiany danych o zdarzeniach zaistniałych dla obiektów z wieloma pojęciami przypadków. Wyznaczone przez autorów cele prezentują się następująco:

- interoperacyjność - zapewnienie standardu OCEL wraz z serializacją z plików formacie JSON/XML ma na celu wsparcie szerokiej gamy języków i systemów
- generalizacja - standard obsługuje przechowywanie zdarzeń, obiektów i ich atrybutów, dodatkowo może zostać rozszerzony
- zapewnienie kolekcji przykładów - przykładowe logi wyodrębnione z systemów informatycznych wspierających procesy biznesowe zostały dostarczone wraz ze standardem
- wsparcie narzędziowe - w celu wsparcia implementacji OCEL w aplikacjach należy zapewnić wsparcie narzędziowe.

Nieformalnym sposobem przedstawienia takiego obiektowego dziennika zdarzeń są dwie tabele, z których jedna reprezentuje rekordy zdarzeń, gdzie każdy wiersz odpowiada odrębnemu zdarzeniu (Rys. 1.1). Druga tabela prezentuje informacje na temat poszczególnych obiektów (Rys. 1.2).

id	activity	timestamp	item	order	package	prepaid-amount	weight	total-weight
e ₁	place order	2020-07-09 08:20:01.527+01:00	{ i ₁ , i ₂ }	{ o ₁ }		200.0		
e ₂	check availability	2020-07-09 08:21:01.527+01:00	{ i ₁ }				10.0	
e ₃	load package	2020-07-09 08:22:01.527+01:00			{ p ₁ }			100.0

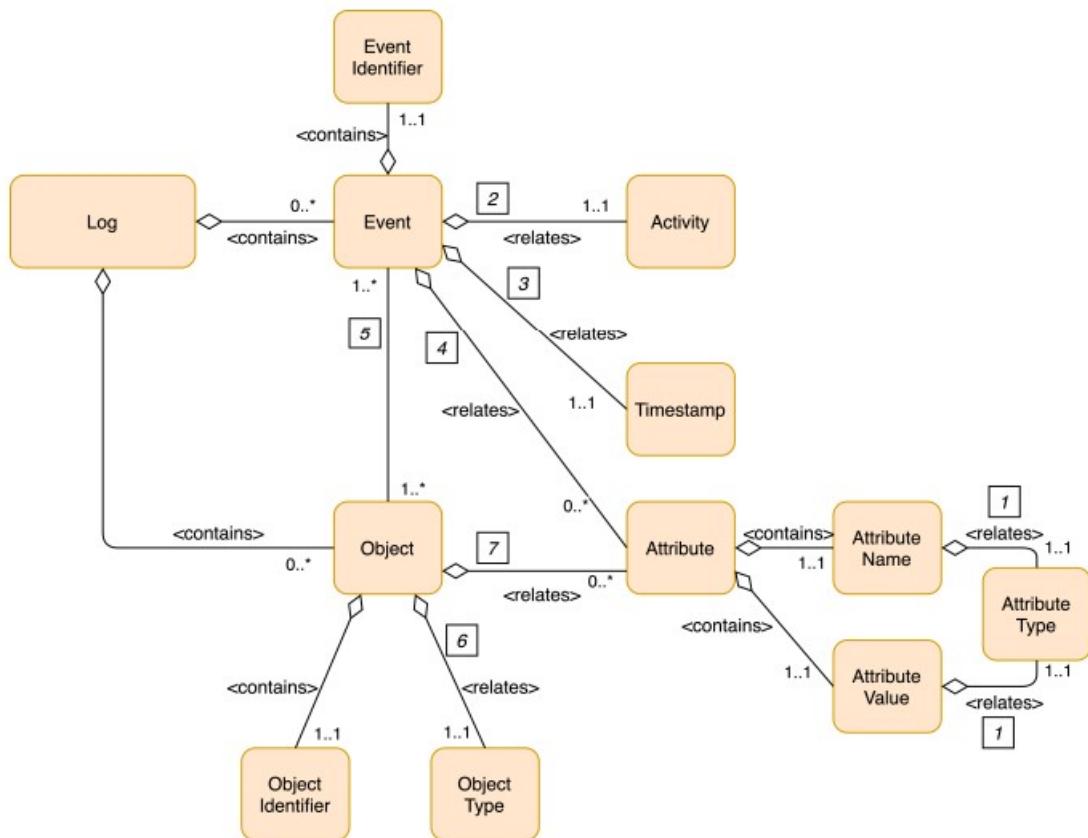
Rys. 1.1. Reprezentacja zdarzeń

Każde zdarzenie reprezentuje zapis wykonania podstawowego procesu biznesowego i jest powiązane z identyfikatorem, aktywnością i znacznikiem czasu. Obiekty reprezentują fizyczne i informacyjne

<u>id</u>	<u>type</u>	<u>customer</u>	<u>costs</u>	<u>color</u>	<u>size</u>
<i>o₁</i>	<i>order</i>	<i>Apple</i>	<i>3500.0</i>		
<i>i₂</i>	<i>item</i>			<i>green</i>	<i>small</i>

Rys. 1.2. Reprezentacja obiektów

jednostki tworzące procesy biznesowe, takie jak materiały, dokumenty, produkty, faktury itp. Każdy obiekt ma identyfikator i jest powiązany z typem obiektu. Dodatkowo każdy obiekt może zawierać wiele atrybutów. Ogólną ideę standardu OCEL zaprezentowano na rysunku 1.3.



Rys. 1.3. Graficzne przedstawienie konceptu dzienników zdarzeń zorientowanych na obiekty

Powyższe informacje oraz grafiki zostały zaczerpnięte ze strony autorów standardu [1], jest ona podstawowym źródłem wiedzy na ten temat. Można tam również znaleźć formalną definicję.

1.1. Cel projektu

Celem projektu jest opracowanie systemu pozwalającego na klasteryzację, filtrowanie i przetwarzanie logów w formacie OCEL. W ramach pracy należy przedstawić autorskie metody filtrowania i

wizualizacji zawartych w dziennikach zdarzeń informacji oraz zaproponować metody klasteryzacji. Do osiągnięcia założonych celów można wykorzystać dostępne biblioteki i narzędzia.

2. Filtrowanie i wizualizacja logów OCEL

W ramach realizacji zdecydowano się na wykorzystanie do implementacji języka Python wraz z użyciem różnych bibliotek, w tym dedykowanej do eksploracji procesów biblioteki pm4py [2]. Prezentacja działania stworzonego systemu odbywa się w ramach notatników Jupyter.

2.1. Wczytywanie logów

Zarówno twórcy w stworzonym repozytorium [3] jak i bibliotek pm4py wspierają wczytywanie logów w formacie jsonocel oraz xmlocel. Wykorzystane dzienniki zdarzeń uzyskano również z ich strony.

```
log_path = "logs/running-example.jsonocel"
ocel = pm4py.read_ocel_json(log_path)
```

Rys. 2.1. Wczytywanie logów

2.2. Filtrowanie i wyświetlanie informacji

Zaimplementowane zostały funkcje pozwalające na filtrowanie i wyświetlanie informacji o wybranych logach w sposób interaktywny dzięki wykorzystaniu biblioteki ipywidgets i pandas.

2.2.1. Podstawowe informacje o logach

Pierwszy z widgetów po podaniu ścieżki do logu pozwala na uzyskanie informacji o:

- ilości różnych obiektów i zdarzeń
- rodzajach aktywności powiązanych z poszczególnymi obiektaми

```

log_path logs/running-example.jsonocel

Your log includes:
- 22367 different events
- 11521 different objects

Object category Number
0      items      20
1      products     17
2      customers   8159
3      orders      2000
4      packages    1325

Activities - products Activities - customers Activities - items
0      reorder item      reorder item      reorder item
1      place order       place order       place order
2      failed delivery   failed delivery   failed delivery
3      pay order         pay order         pay order
4      confirm order     confirm order     confirm order

Activities - orders Activities - packages
0      reorder item      failed delivery
1      place order       package delivered
2      failed delivery   create package
3      pay order         send package
4      confirm order     NaN

```

Rys. 2.2. Wyświetlanie podstawowych informacji o logu

2.2.2. Filtracja po id zdarzenia i rodzaju obiektu

Drugi z widgetów pozwala na uzyskanie informacji o tym co w zdarzeniu o wybranym id działało się z obiekttami danego typu:

<input style="width: 100px; height: 20px; border: 1px solid black; padding: 2px; margin-right: 10px;" type="text" value="1.0"/>	<input style="width: 100px; height: 20px; border: 1px solid black; padding: 2px; margin-right: 10px;" type="text" value="products"/>					
Filter by event id and object type						
ocel:id ocel:activity ocel:timestamp ocel:oid ocel:type ocel:qualifier						
1	1.0	place order	2019-05-20 09:07:47	Echo Studio	products	None
7	1.0	place order	2019-05-20 09:07:47	Echo Show 8	products	None
8	1.0	place order	2019-05-20 09:07:47	Fire Stick 4K	products	None
9	1.0	place order	2019-05-20 09:07:47	Echo	products	None

Rys. 2.3. Filtracja po id zdarzenia i rodzaju obiektu

2.2.3. Wyświetlanie zdarzeń powiązanych z wybranym klientem

Trzeci z widgetów pozwala na uzyskanie informacji o wszystkich zdarzeniach, które powiązane są z klientem, którego personalia można wybrać z listy rozwijanej spośród wszystkich występujących w danym logu.

Filter by event customer name						
	ocel:id	ocel:activity	ocel:timestamp	ocel:oid	ocel:type	ocel:qualifier
2	1.0	place order	2019-05-20 09:07:47	Marco Pegoraro	customers	None
26	4.0	confirm order	2019-05-20 11:13:54	Marco Pegoraro	customers	None
35	5.0	pick item	2019-05-20 11:20:13	Marco Pegoraro	customers	None
59	8.0	item out of stock	2019-05-20 13:54:37	Marco Pegoraro	customers	None
165	26.0	reorder item	2019-05-21 10:03:49	Marco Pegoraro	customers	None
...
181031	22233.0	payment reminder	2020-05-27 08:54:09	Marco Pegoraro	customers	None
181563	22295.0	pay order	2020-06-05 17:32:43	Marco Pegoraro	customers	None
182076	22340.0	create package	2020-07-13 01:00:00	Marco Pegoraro	customers	None
182084	22341.0	send package	2020-07-13 14:05:37	Marco Pegoraro	customers	None
182092	22342.0	package delivered	2020-07-14 17:02:12	Marco Pegoraro	customers	None

Rys. 2.4. Zdarzenia powiązane z wybranym klientem

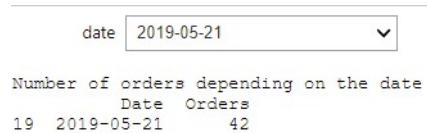
Kolejny z widgetów pozwala na uzyskanie informacji o zdarzeniach powiązanych z wybranym typem aktywności i obiektu, przykładowo możemy zweryfikować zamówienia, które nie zostały dostarczone.

Filter by object type and activity						
	ocel:id	ocel:activity	ocel:timestamp	ocel:oid	ocel:type	ocel:qualifier
958	142.0	failed delivery	2019-05-23 14:48:41	990011	orders	None
964	142.0	failed delivery	2019-05-23 14:48:41	990002	orders	None
965	142.0	failed delivery	2019-05-23 14:48:41	990013	orders	None
966	142.0	failed delivery	2019-05-23 14:48:41	990012	orders	None
975	142.0	failed delivery	2019-05-23 14:48:41	990009	orders	None
...
181461	22283.0	failed delivery	2020-06-04 10:59:59	991942	orders	None
181922	22321.0	failed delivery	2020-06-23 12:02:31	991924	orders	None
181925	22321.0	failed delivery	2020-06-23 12:02:31	991983	orders	None
182039	22334.0	failed delivery	2020-07-03 14:38:32	991960	orders	None
182140	22348.0	failed delivery	2020-07-24 16:15:58	991939	orders	None

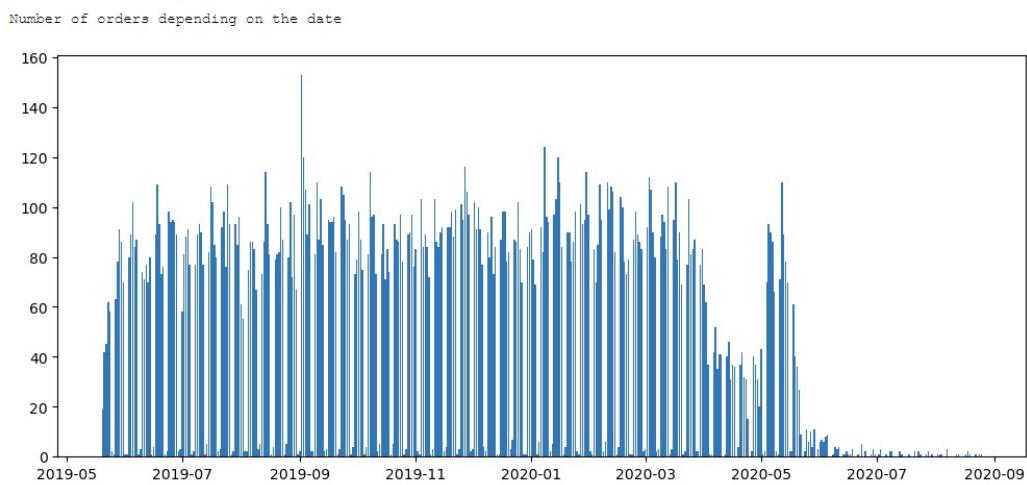
Rys. 2.5. Zdarzenia o zadanej aktywności powiązane z wybranym typem obiektu

2.2.4. Sprawdzenie ilości zamówień w danym dniu

Jeden ze stworzonych widgetów pozwala na sprawdzenie ilości złożonych zamówień w danym dniu, dodatkowo wykonany został wykres zależności ilości zamówień od czasu.



Rys. 2.6. Widget pozwalający na sprawdzenie ilości zamówień z danego dnia



Rys. 2.7. Wykres pozwalający na śledzenie ilości zamówień na przestrzeni czasu

2.2.5. Podgląd interakcji wybranego obiektu z innymi obiektami

Zaimplementowano możliwość wyświetlenia interakcji obiektu o wybranym oid, w którym każdy wiersz odpowiada innej kombinacji (zdarzenie, wybrany obiekt, inny powiązany obiekt). Uwzględniane są ich typy oraz łączące je zdarzenie.

oid 880001

Chosen object interaction summary:						
	ocel: eid	ocel: activity	ocel: oid	ocel: type	ocel: oid_2	ocel: type_2
0	1.0	place order	880001	items	Echo Studio	products
1	1.0	place order	880001	items	Marco Pegoraro	customers
2	1.0	place order	880001	items	880004	items
3	1.0	place order	880001	items	880003	items
4	1.0	place order	880001	items	880002	items
...
2131685	988.0	send package	880001	items	iPad Pro	products
2131686	988.0	send package	880001	items	880262	items
2131687	988.0	send package	880001	items	880391	items
2131688	988.0	send package	880001	items	990075	orders
2131689	988.0	send package	880001	items	880264	items

Rys. 2.8. Interakcje między obiektami

2.2.6. Wyświetlanie obiektów powiązanych z wybranym obiektem

Kolejny stworzony widget pozwala na podgląd ilości powiązanych ze zdarzeniem o wybranym eid.

eid

```
Related objects for chosen event: Counter({'items': 4, 'products': 4, 'customers': 1, 'orders': 1})
```

Rys. 2.9. Podgląd obiektów powiązanych z wybranym zdarzeniem

2.2.7. Podgląd zakończenia cyklu życia obiektu

Stworzono także widget pozwalający na podgląd kiedy dany obiekt zakończył swój cykl życia jednak ze względu na jego wielkość nie został zamieszczony w niniejszej pracy. Zawiera on między innymi informacje na temat czasu, aktywności, wag, ceny, powiązanych produktów czy klienta.

2.3. Testy przeprowadzone dla innego logu

W powyższej części pracy przedstawiono uzyskane wizualizacje dla logu 'running-example.jsonocel', który można znaleźć w załączonym repozytorium kodu. Poniższe rezultaty obrazują wykorzystanie stworzonego rozwiązania dla logu "recruiting.jsonocel", zawiera on informacje o prowadzonych przez firmę procesach rekrutacyjnych.

log_path

```
Your log includes:
- 6980 different events
- 1505 different objects

Object category Number
0    vacancies      140
1    managers        6
2    applicants      288
3    applications    916
4    recruiters      20

Activities - vacancies Activities - managers Activities - applicants \
0    change manager   change manager   submit application
1    submit application consult manager send rejection
2    open vacancy     open vacancy    job offer declined
3    close vacancy (no hire) make job offer offer accepted and hired
4    assign vacancy    conduct interview invite for interview

Activities - applications Activities - recruiters \
0    submit application consult manager
1    consult manager    assign recruiter
2    assign recruiter   send rejection
3    send rejection    job offer declined
4    assign vacancy     offer accepted and hired

Activities - offers
0    job offer declined
1    make job offer
2    offer accepted and hired
```

Rys. 2.10. Wyświetlanie podstawowych informacji o logu "recruiting.jsonocel"

Filter events by manager name

ocel:eid	ocel:activity	ocel:timestamp	ocel:oid	ocel:type	ocel:qualifier	
211	77	open vacancy	2019-05-23 14:03:33+00:00	Alexander Rinke	managers	None
308	114	open vacancy	2019-05-26 12:01:03+00:00	Alexander Rinke	managers	None
444	162	open vacancy	2019-05-28 12:09:34+00:00	Alexander Rinke	managers	None
600	219	open vacancy	2019-05-30 03:32:51+00:00	Alexander Rinke	managers	None
1247	455	open vacancy	2019-06-06 05:19:20+00:00	Alexander Rinke	managers	None
...	
21881	6920	change manager	2020-05-12 11:19:17+00:00	Alexander Rinke	managers	None
21887	6922	change manager	2020-05-21 13:45:57+00:00	Alexander Rinke	managers	None
21897	6929	change manager	2020-06-05 18:03:31+00:00	Alexander Rinke	managers	None
21917	6946	change manager	2020-06-30 13:30:45+00:00	Alexander Rinke	managers	None
21944	6966	change manager	2020-07-30 10:30:08+00:00	Alexander Rinke	managers	None

Rys. 2.11. Filtracja zdarzeń powiązanych z wybranym managerem

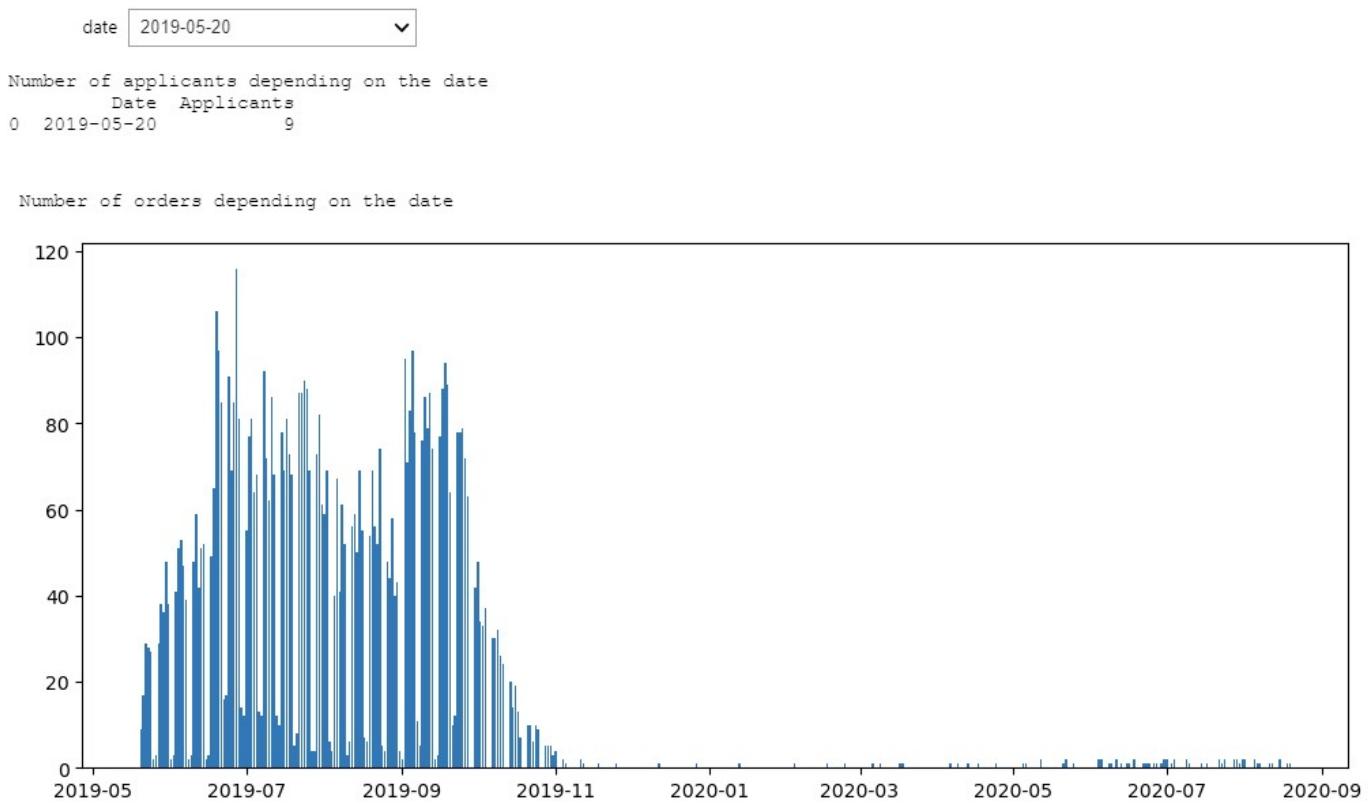
Object type:

Activity:

Filter by object type and activity

ocel:eid	ocel:activity	ocel:timestamp	ocel:oid	ocel:type	ocel:qualifier	
3093	1102	make job offer	2019-06-20 15:15:56+00:00	Sue Johnson	applicants	None
3539	1246	make job offer	2019-06-24 07:33:47+00:00	Stephan Jones	applicants	None
3722	1300	make job offer	2019-06-24 13:59:28+00:00	Vanessa Lemmens	applicants	None
3927	1365	make job offer	2019-06-25 12:15:55+00:00	Chun Jansen	applicants	None
4738	1638	make job offer	2019-06-28 12:17:17+00:00	Vanessa Li	applicants	None
...	
21222	6737	make job offer	2019-10-11 08:08:50+00:00	Chun Lemmens	applicants	None
21301	6758	make job offer	2019-10-14 08:50:44+00:00	Sonia Weber	applicants	None
21389	6783	make job offer	2019-10-15 10:08:06+00:00	Pete Zimmermann	applicants	None
21560	6831	make job offer	2019-10-21 12:25:05+00:00	Andre Weber	applicants	None
21616	6846	make job offer	2019-10-22 14:23:36+00:00	Mary Jansen	applicants	None

Rys. 2.12. Zdarzenia o zadanej aktywności powiązane z wybranym typem obiektu



Rys. 2.13. Widget wraz z wykresem pozwalające na sprawdzenie ilości aplikacji w wybranym dniu

eid

```
Related objects for chosen event: Counter({'applicants': 1, 'applications': 1, 'vacancies': 1})
```

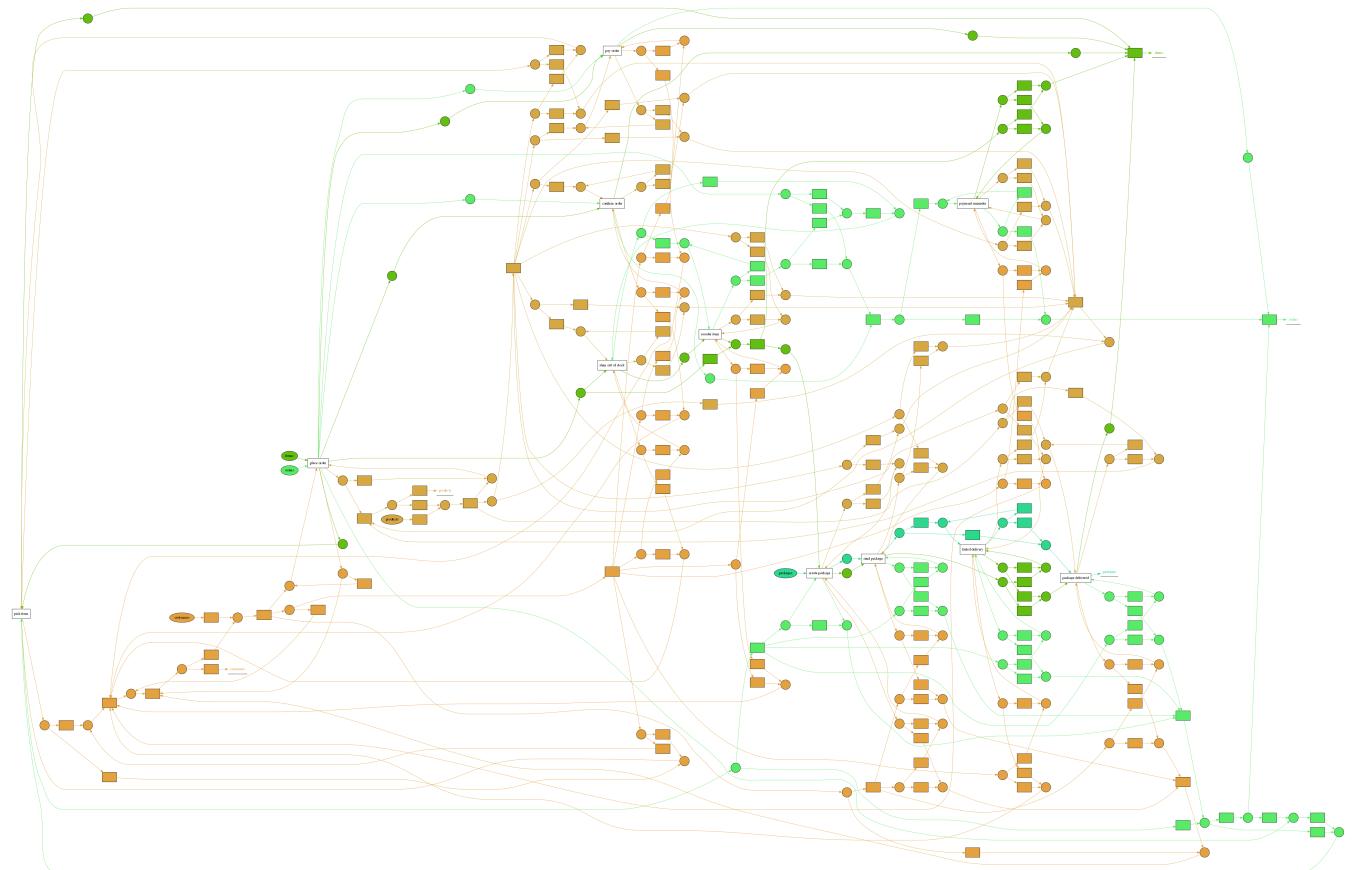
Rys. 2.14. Podgląd obiektów powiązanych z wybranym zdarzeniem

obj_type	vacancies	▼
Chosen objects type termination:		
ocel:eid	ocel:timestamp	ocel:activity
0 1670	2019-06-28 15:53:46+00:00	conduct interview
1 2179	2019-07-09 08:23:47+00:00	conduct interview
2 2638	2019-07-17 08:49:37+00:00	conduct interview
3 2641	2019-07-17 09:23:17+00:00	conduct interview
4 2999	2019-07-23 13:08:29+00:00	offer accepted and hired
..
135 6976	2020-08-12 18:43:18+00:00	close vacancy (no hire)
136 6977	2020-08-15 03:49:22+00:00	close vacancy (no hire)
137 6978	2020-08-15 13:45:15+00:00	close vacancy (no hire)
138 6979	2020-08-18 11:13:20+00:00	close vacancy (no hire)
139 6980	2020-08-19 20:53:18+00:00	close vacancy (no hire)
ocel:type:applicants ocel:type:applications \		
0 [Ray Jansen]	[Application[770044]]	
1 [Toon Weber]	[Application[770046]]	
2 [Andre Schmidt]	[Application[770152]]	
3 [Andre Wagner]	[Application[770116]]	
4 [Sonia Wang]	[Application[770213]]	
..
135 NaN		NaN
136 NaN		NaN
137 NaN		NaN
138 NaN		NaN
139 NaN		NaN
ocel:type:vacancies ocel:type:recruiters \		
0 [Vacancy[550004] - Programmer]		[Jana Kershaw]
1 [Vacancy[550002] - Consultant]		[Dionne Meister]
2 [Vacancy[550011] - Programmer]		[Simon Meister]
3 [Vacancy[550019] - Manager]		[Dionne Meister]
4 [Vacancy[550020] - Consultant]		[Dionne Keane]
..
135 [Vacancy[550123] - Manager]		NaN
136 [Vacancy[550127] - Programmer]		NaN
137 [Vacancy[550128] - Programmer]		NaN
138 [Vacancy[550130] - Consultant]		NaN
139 [Vacancy[550133] - Consultant]		NaN
ocel:type:managers ocel:type:offers		
0 [Martin Klenk]		NaN
1 [Bastian Nominacher]		NaN
2 [Miguel Milano]		NaN
3 [Vaishnavi Sashikanth]		NaN
4 NaN	[Offer[[990019]]]	
..
135 NaN		NaN
136 NaN		NaN
137 NaN		NaN
138 NaN		NaN
139 NaN		NaN

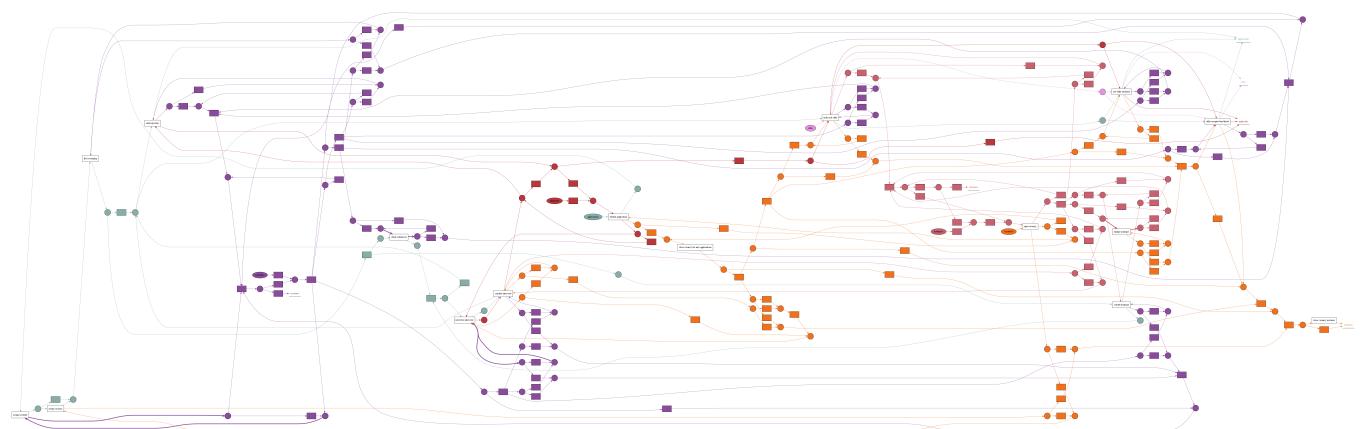
Rys. 2.15. Podgląd momentu, kiedy obiekty danego typu kończą cykl życia

2.4. Wizualizacja logów w postaci sieci Petriego

Biblioteka pm4py zawiera funkcję, która pozwala na odkrywanie obiektocentycznej sieci Petriego na podstawie dostarczonego logu OCEL, więcej na temat eksploracji procesów na bazie logów obiektowych można znaleźć w pracy Juliana Kofferatha [4].



Rys. 2.16. Sieć Petriego powstała na bazie logu "running-example.jsonocel"



Rys. 2.17. Sieć Petriego powstała na bazie logu "recruiting.jsonocel"

3. Klasteryzacja logów OCEL

3.1. Rozróżnianie logów

Jako metrykę stosowaną do rozróżniania eventów w klastrze wybrano rodzaj obiektów powiązanych z danym eventem. Eventy zostały przekształcone do postaci wektorowej w sposób opisany poniżej. Zakładając, że dany log ma zdefiniowaną następującą listę obiektów [o1, o2, o3, o4, o5, o6] to event powiązany z obiektami ‘o1’, ‘o4’ i ‘o6’ będzie reprezentowany przez następujący wektor [1, 0, 0, 1, 0, 1]. Liczba ‘1’ odpowiada obecności obiektu o danym indeksie w evencie, a liczba ‘0’ braku braku takiej obecności.

3.2. Algorytm klasteryzacji

W celu automatycznego odkrycia eventów z takimi samymi obiektami zastosowany został algorytm DBSCAN.

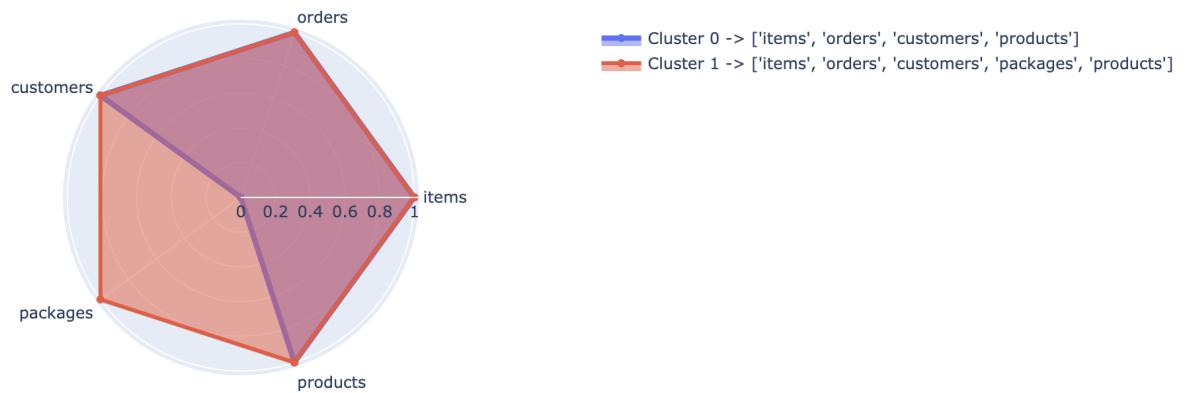
3.3. Wizualizacja klastrów

W celu porównania powstały klastry zastosowano tak zwany spider plot. W celu zachowania czytelności wykresu, na jednym plocie przedstawiono maksymalnie trzy klastry.

3.3.1. Order Management log

Log pochodzi ze strony <https://ocel-standard.org/>

Clusters comparison

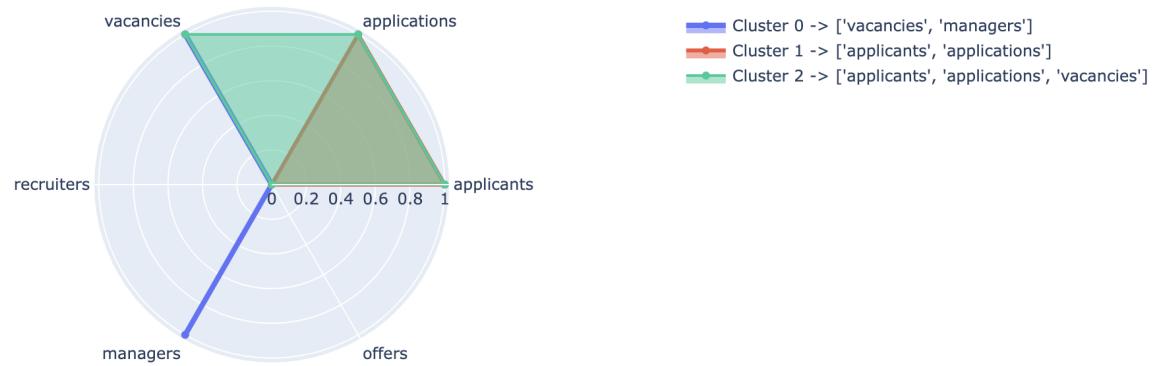


Rys. 3.1. Klastry w logu ‘Order Management log’

3.3.2. Recruiting Process (Simulated)

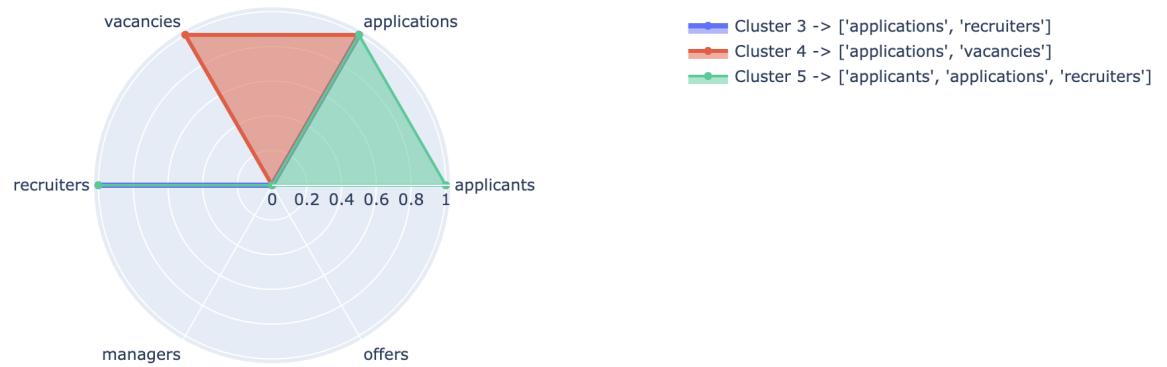
Log pochodzi ze strony <https://ocel-standard.org/>

Clusters comparison



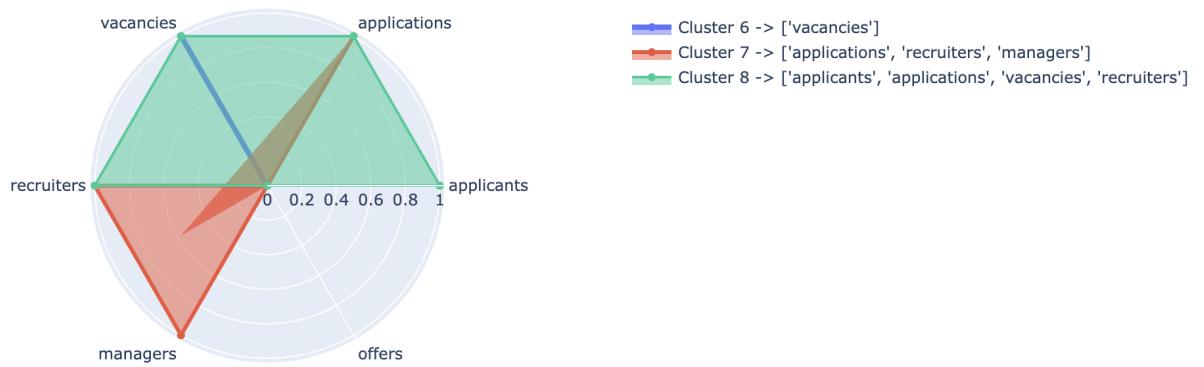
Rys. 3.2. Klastry w logu ‘Recruiting Process (Simulated)’

Clusters comparison



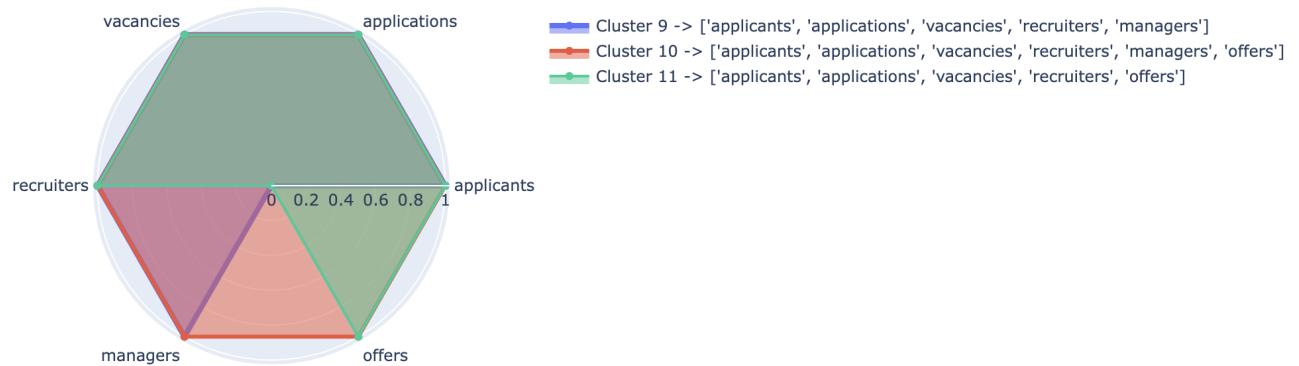
Rys. 3.3. Klastry w logu ‘Recruiting Process (Simulated)‘

Clusters comparison



Rys. 3.4. Klastry w logu ‘Recruiting Process (Simulated)‘

Clusters comparison

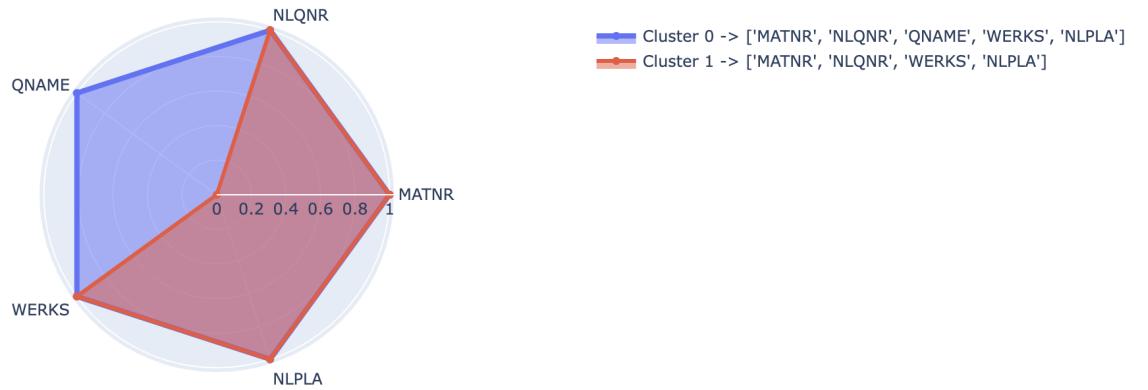


Rys. 3.5. Klastry w logu ‘Recruiting Process (Simulated)‘

3.3.3. SAP ERP IDES instance - Transfer Order (Storage)

Log pochodzi ze strony <https://ocel-standard.org/>

Clusters comparison



Rys. 3.6. Klastry w logu ‘SAP ERP IDES instance - Transfer Order (Storage)‘

4. Podsumowanie

W ramach niniejszej pracy dokonaliśmy przeglądu dostępnych źródeł dotyczących logów obiektocentrycznych. Zgodnie z tematem projektu zaimplementowaliśmy ich filtrowanie oraz dokonaliśmy ich wizualizacji w postaci sieci Petriego. Udało nam się również dokonać klasteryzacji zdarzeń biorąc pod uwagę obiekty, które brały w nim udział.

Link do repozytorium projektu: <https://github.com/wojciechzyla/projekt-mb.git>

Bibliografia

- [1] RWTH Aachen University. *OCEL STANDARD*. 2023. URL: <https://ocel-standard.org/> (term. wiz. 2023-06-12).
- [2] Process Mining Group of the Fraunhofer Institute for Applied Information Technology. *PM4PY*. 2023. URL: <https://pm4py.fit.fraunhofer.de/> (term. wiz. 2023-06-12).
- [3] Process Mining Group of the Fraunhofer Institute for Applied Information Technology. *ocel-support*. 2023. URL: <https://github.com/OCEL-standard/ocel-support> (term. wiz. 2023-06-12).
- [4] Julian Kofferath. *Applying Object-Centric Process Mining to a Real Business Process: Inventory in SAP ERP Systems*. Aachen: Process i Data Science (PADS) Chair, RWTH Aachen University, 2022.