

# Process Mining Data

## ▼ Overview

Process mining is a powerful tool that allows organizations to analyze their business processes based on event logs

### Where

- **Enterprise Systems:** Process mining data is primarily extracted from enterprise systems like ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), and BPM (Business Process Management) systems. Examples include SAP, Oracle, and Salesforce.
- **IT Systems:** IT systems generate logs that can be mined, including application servers, databases, and web services.
- **Healthcare Systems:** Hospital information systems and electronic health records provide event logs for process mining.
- **Manufacturing:** Manufacturing execution systems (MES) and other operational systems log data on production processes.

### Critiques

- **Data Quality:** The quality of event logs can vary, impacting the accuracy of the process models.
- **Privacy Concerns:** Handling sensitive data requires stringent privacy and compliance measures.
- **Complexity:** Real-world processes can be highly complex, making it difficult to create accurate models.
- **Resource Intensive:** Process mining can be computationally intensive and require significant resources.
- **Interpretation Challenges:** Interpreting the results of process mining requires expertise and can be subjective.

### Importance

- **Optimization:** Helps identify bottlenecks and inefficiencies in processes, enabling organizations to optimize operations.
- **Compliance:** Ensures processes comply with regulations and standards.
- **Transparency:** Provides a clear view of actual process flows versus intended processes.
- **Continuous Improvement:** Supports continuous process improvement initiatives by providing actionable insights.
- **Cost Reduction:** By optimizing processes, organizations can reduce operational costs and improve profitability.

### Extraction - Exploration & Interpretation

- **Extraction:** Data is extracted from event logs, which include information about cases (process instances), activities (tasks or steps), and timestamps.
- **Exploration:** Analyzing the process model to understand the flow, detect patterns, and identify deviations from the expected process.
- **Interpretation:** Drawing insights from the process model, including identifying bottlenecks, inefficiencies, and opportunities for improvement.

### Real World Example of Event Data - Logs

- **Retail:** A large retail chain collects data on customer transactions, including the steps from order placement to delivery. Event logs include timestamps for order received, payment processed, item picked, packed, shipped, and delivered.
- **Healthcare:** A hospital system logs patient treatment processes, including registration, consultation, diagnosis, treatment, and discharge. Event logs capture timestamps for each step, allowing for analysis of patient flow and treatment efficiency.
- **Banking:** A bank's loan approval process logs each step, from application submission, credit check, underwriting, approval, and disbursement. Event logs help analyze the time taken at each step and identify delays.
- **Manufacturing:** A factory logs production processes, including raw material received, production started, quality checks, packaging, and shipping. Event logs help identify production delays and quality issues.

### Detailed Example: Healthcare Process Mining

- **Where:** Hospital information systems (HIS) and electronic health records (EHR).
- **Critiques:** Data quality issues due to inconsistent recording practices; privacy concerns related to patient data.
- **Importance:** Helps improve patient care by optimizing treatment processes and reducing wait times.
- **Extraction:** Collecting event logs from HIS and EHR, including patient check-in, consultations, lab tests, treatments, and discharge.
- **Exploration:** Analyzing the flow of patients through different departments and identifying bottlenecks.

- **Interpretation:** Drawing insights to streamline patient flow, reduce wait times, and improve overall care quality.

Process mining provides a data-driven approach to understanding and improving business processes across various industries, offering significant benefits while also posing certain challenges.

▼ **Event Data/log**

▼ **Traditional Event Data**

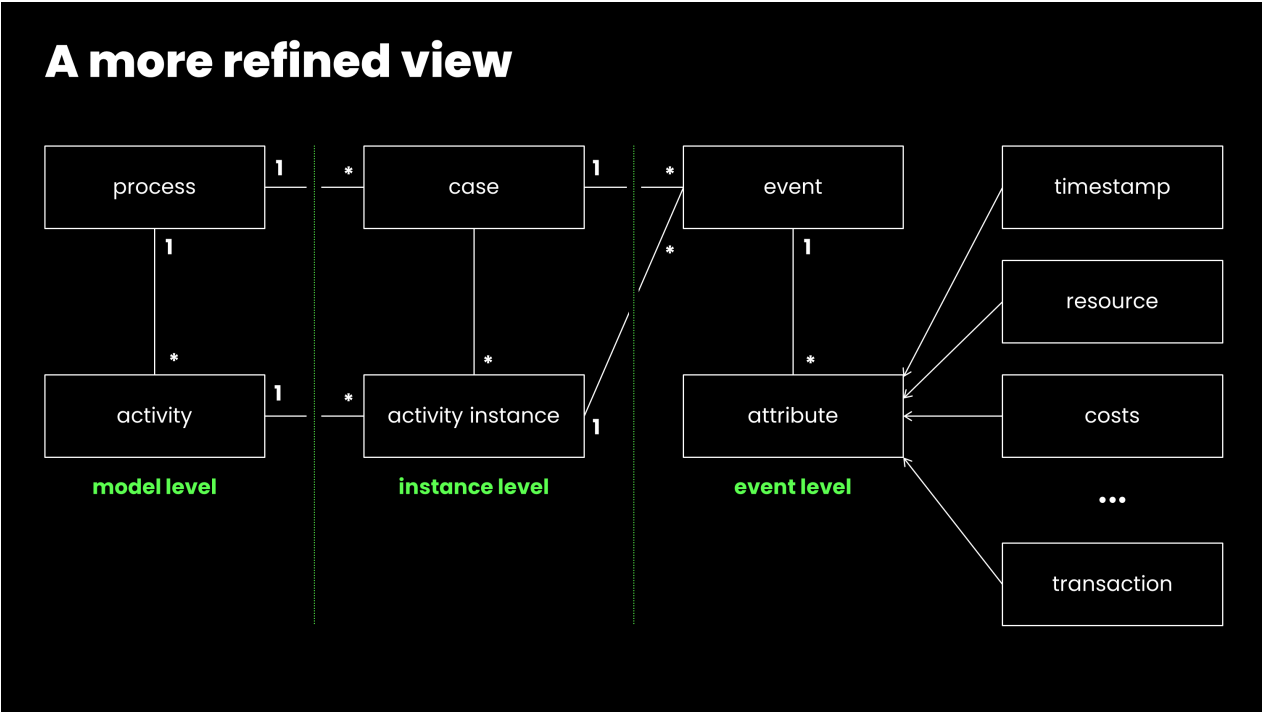
Event data need to be *extracted* from information systems used to support the processes that need to be analyzed. Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), and Supply Chain Management (SCM) systems store events. Consider, for example, the *Purchase-to-Pay* (P2P) and *Order-to-Cash* (O2C) processes that most organizations have. The P2P process is there to input products and services from suppliers into the organization. The O2C process is there to output products and services to customers (including invoicing and handling payments). These processes are supported by systems like SAP S/4HANA, Oracle E-Business Suite, and Microsoft Dynamics 365. These systems have many database tables storing information about these and other operational processes. This means that events are *scattered over many different tables* and often also over *different systems*. Also, when systems are domain-specific (e.g., healthcare) or organization-specific (i.e., homebrew software), there are typically many database tables containing events. Note that most database tables managed by contemporary information systems contain dates or timestamps. Some systems also store changes in these tables (e.g., the so-called "change tables" in SAP).

Hence, there are plenty of events. However, the challenge is to extract and convert these into a format that can be used for process mining. This is an integral part of any process mining effort and typically be time-consuming. In the simplest setting, events are represented by a *case identifier*, an *activity name*, and a *timestamp*. There may be optional attributes like resource, location, cost, etc.

Remember: Event data

Case ID	Activity	Resource	Timestamp	Product	Prod-price	Quantity	Address
...	...	...	...	...	...	...	...
6350	place order	Aiden	2018/02/13 14:29:45.000	APPLE iPhone 6 16 GB	639,00 €	5	NL-7751DG-21
6283	pay	Lily	2018/02/13 14:39:25.000	SAMSUNG Galaxy S6 32 GB	543,99 €	3	NL-7828AM-11a
6253	prepare delivery	Sophia	2018/02/13 15:01:33.000	APPLE iPhone 6s 16 GB	599,00 €	3	NL-7887AC-13
6257	prepare delivery	Aiden	2018/02/13 15:03:43.000	SAMSUNG Galaxy S6 32 GB	543,99 €	1	NL-9521KJ-34
6185	confirm payment	Emily	2018/02/13 15:05:36.000	SAMSUNG Galaxy S4	329,00 €	1	NL-9521GC-32
6218	confirm payment	Emily	2018/02/13 15:08:11.000	APPLE iPhone 6s 16 GB	969,00 €	2	NL-7948BX-10
6245	make delivery	Michael	2018/02/13 15:14:04.000	APPLE iPhone 6 16 GB	639,00 €	3	NL-7905AX-38
6272	pay	Emily	2018/02/13 15:20:36.000	APPLE iPhone 6 16 GB	639,00 €	1	NL-7821AC-3
6269	pay	Charlotte	2018/02/13 15:25:21.000	SAMSUNG Galaxy S4	329,00 €	1	NL-7907EJ-42
6212	prepare delivery	Sophia	2018/02/13 15:43:39.000	HUAWEI P8	199,00 €	1	NL-7905AX-38
6323	send invoice	Alexander	2018/02/13 15:46:08.000	APPLE iPhone 6s 16 GB	969,00 €	1	NL-7833HT-15
6246	confirm payment	Jack	2018/02/13 15:56:03.000	SAMSUNG Galaxy S4	329,00 €	3	NL-7833HT-15
6347	send invoice	Jack	2018/02/13 15:57:42.000	SAMSUNG Galaxy S4	329,00 €	3	NL-7905AX-38
6351	place order	Zoe	2018/02/13 16:17:37.000	APPLE iPhone 6s 16 GB	969,00 €	3	NL-9521GC-32
6204	prepare delivery	Sophia	2018/02/13 16:31:28.000	SAMSUNG Core Prime G361	135,00 €	1	NL-7828AM-11a
6204	make delivery	Kaylee	2018/02/13 16:51:54.000	SAMSUNG Core Prime G361	135,00 €	1	NL-7828AM-11a
6265	confirm payment	Lily	2018/02/13 16:55:55.000	SAMSUNG Galaxy S4	329,00 €	4	NL-9521GC-32
6250	confirm payment	Jack	2018/02/13 17:03:26.000	MOTO G	199,00 €	4	NL-7942GT-2
6328	send invoice	Lily	2018/02/13 17:30:16.000	APPLE iPhone 6s 64 GB	858,00 €	4	NL-9514BV-16
6352	place order	Aiden	2018/02/13 17:53:22.000	APPLE iPhone 6 16 GB	639,00 €	2	NL-9514BV-16
6317	send invoice	Jack	2018/02/13 18:45:30.000	APPLE iPhone 6s 64 GB	858,00 €	5	NL-7907EJ-42
6353	place order	Sophia	2018/02/13 20:16:20.000	APPLE iPhone 5s 16 GB	449,00 €	4	NL-7751AR-19
...	...	...	...	...	...	...	...

For process discovery and conformance checking, we focussed on control flow, i.e., the ordering of activities. However, in reality, there are much more data.



Events may have any number of attributes. However, in the classical setting, events have at least a timestamp, refer to an activity, and belong to a case. Sometimes we use the notion of an *activity instance*, i.e., the execution of an activity for a specific case. One activity instance may consist of several events, e.g., the start and completion of the activity instance. Therefore, an event may

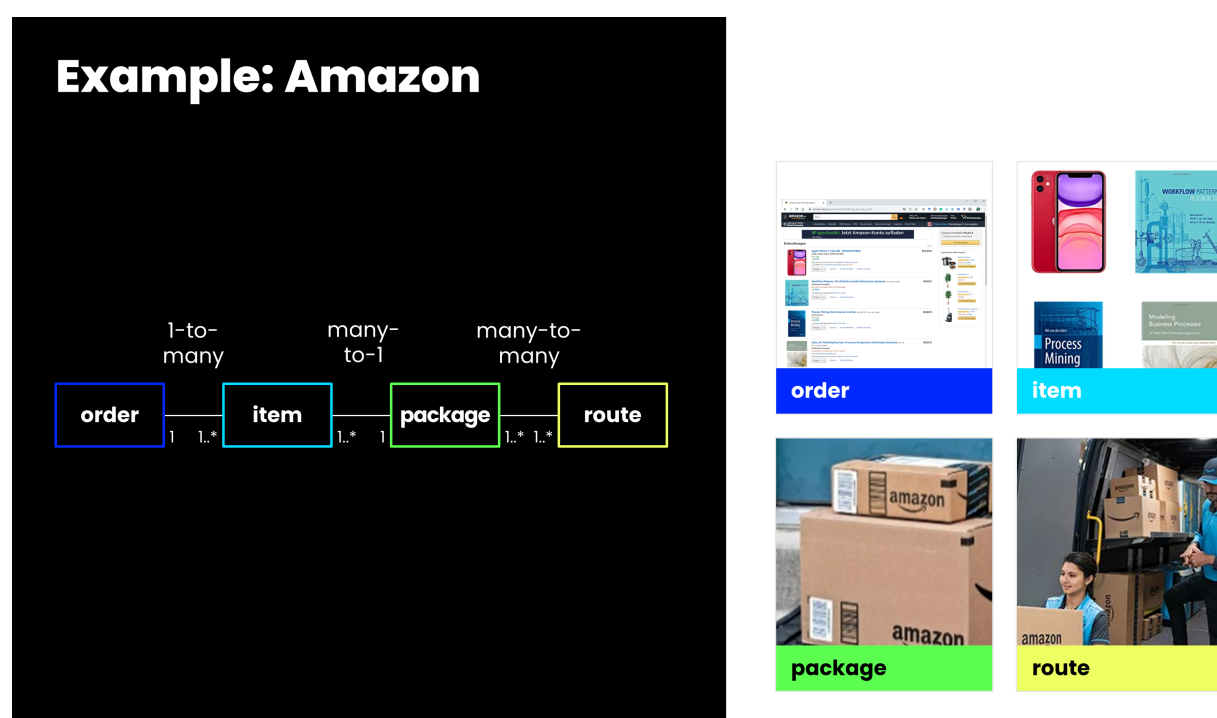
have a transaction attribute. Also, cases may have attributes (not shown) and belong to a process. Traditionally, we assume that case attributes do not change over time (e.g., the birth date of a patient) and that a case belongs to precisely one process.

XES (eXtensible Event Stream) is the IEEE standard for storing traditional event data, as described above. The IEEE Task Force on Process Mining started to work on this in September 2010, and it became an official IEEE standard in 2016. XES supports the standard notions, including activity instances. Classifiers are used to attach labels to events. There is always at least one classifier, and by default, this is the activity name. However, it is also possible to project events onto resources, locations, departments, etc., or combinations of attributes. To handle activities that take time, XES provides the possibility to represent lifecycle information and connect events through activity instances. As mentioned, an activity instance is a collection of related events that together represent the execution of an activity for a case. The XES lifecycle model distinguishes between the following types of events: schedule, assign, withdraw, reassign, start, suspend, resume, abort, complete, autoskip, and manualskip. If event data contain such lifecycle information, it is possible to measure service times, waiting times, synchronization times, idle times, etc.

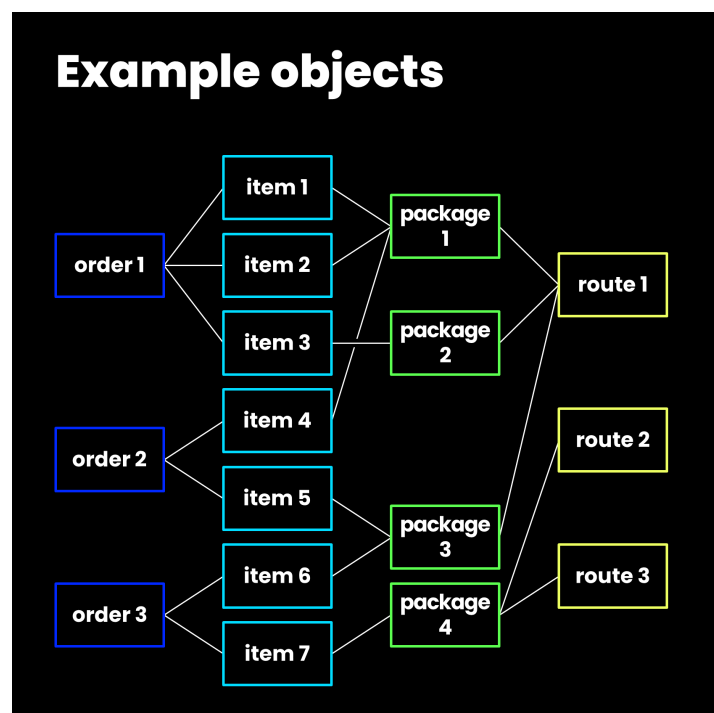
Although the corresponding traditional event logs (stored using XES or not) provide a clear starting point for process mining, they also impose a few limitations. When extracting the event log, one is forced to pick a *viewpoint* in the form of a *case identifier*. This means data extraction needs to be done for every view. Also, events cannot refer to multiple objects. This complicates the extraction of event data from source systems.

## ▼ Data Extraction: An Example

To illustrate the challenges related to extracting data, we consider ordering products from an online shop like Amazon. Customers place orders that may consist of multiple items. Ordered items are delivered in packages. One package may contain multiple items, which may originate from different orders. Also, the items in one order may be distributed over multiple packages, because of availability, weight, size, location, etc. Packages are delivered on routes. However, customers may not be at home, and the package cannot be delivered. As a result, the same package may be on multiple routes.



The relation between orders and items is a *one-to-many* relation, i.e., an order may contain multiple items, but each item is part of one order. The relation between packages and items is also a *one-to-many* relation, i.e., a package may contain multiple items, but each item is part of one package. The indirect relation between orders and packages is a *many-to-many* relation, because items in one order may be scattered over multiple packages, and one package may contain items of multiple orders. In the example below, items from the first order end up in two packages, and items in the first package originate from two orders. Looking at the database schema of a typical information system reveals that one-to-many and many-to-many relations are the rule and not the exception. Moreover, events refer to multiple objects. For example, consider the event of a customer placing an order consisting of three items. This event refers to five objects: one customer object, one order object, and three item objects.



Items of order 1 end up in packages 1 and 2

Package 1 contains items of orders 1 and 2, etc.

## What is a suitable case notion?

Because of the one-to-many and many-to-many relations between objects and events referring to multiple objects, it is often not clear what case notion to pick. In the example, we can pick orders as case notion. Alternatively, we can pick items or packages as a case notion. However, the discovered process models look completely different. Also, the cardinality of events changes. Placing an order consisting of three items is one event at the order level, but when using items as a case notion, there must be three events. Delivering a package consisting of five items is one event at the package level, but when using items as a case notion, there must be five delivery events. Therefore, the choice of case notion has a dramatic effect on both the event log and the discovered process models.

These examples show that it is far from trivial to extract event data.

### ▼ Exercise

#### Question 1

1.0/1.0 point (ungraded)

Select the correct statements about event data.

☒ An event may have many attributes.

☐ Transactional information is a must when recording event data.

☐ One activity instance is only associated with one event.

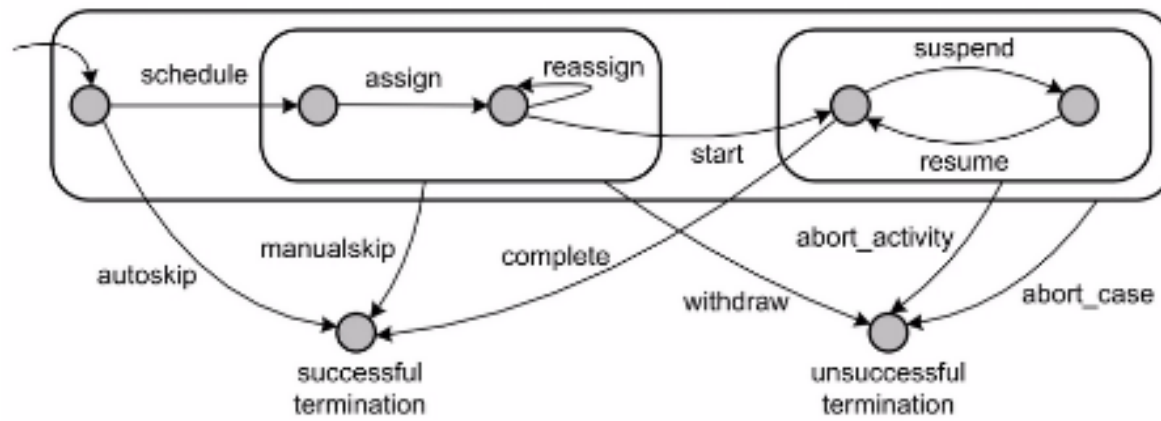
☒ An event should at least contain a case identifier, an activity name, and a timestamp



## Question 2

1/1 point (ungraded)

According to the transactional model for activities, which types represent successful terminations?



☒ autoskip

☒ complete

☐ abort\_activity

☐ withdraw

☒ manualskip

☐ abort\_case

☐ schedule

☐ assign

☐ suspend

☐ resume




## ▼ Exploring Event Data and Extraction Challenges

### ▼ The Need for Object-Centric Event Data

- **Traditional Process Mining:**
  - Requires selecting a single case identifier per event.
  - Describes processes by isolating cases and examining one type of object at a time.
- **Object-Centric Process Mining (OCPM):**
  - Starts with event data where each event can refer to multiple objects.
  - Applicable in processes like Purchase-to-Pay (P2P) and Order-to-Cash (O2C).
  - Involves multiple related objects such as purchase orders, items, suppliers, etc.
  - Events may refer to different objects and multiple objects of the same type.
- **Challenges with Traditional Process Mining:**

- Forces transformation of data to refer to a single case identifier.
- Requires repeated data extraction and transformation for different perspectives.
- Loses relationships between objects and events, creating a two-dimensional view of a three-dimensional reality.
- May obscure root causes of performance and compliance issues.
- **Benefits of OCPM:**
  - Provides a more comprehensive and accurate view of processes.
  - Helps uncover hidden issues by maintaining relationships between objects and events.
- **Resources:**
  - Explore the OCEL 2.0 standard at [OCEL Standard](https://ocel-standard.org/).
  - Read the tutorial-style paper "Object-Centric Process Mining: Unraveling the Fabric of Real Processes".

▼ **Object-Centric Process Mining**

<https://ocel-standard.org/>

Object-Centric Process Mining (OCPM) handles event data where each event can refer to multiple objects, capturing the complex interactions between them. For example, in a sales process, a single event may involve multiple sales orders and items, maintaining their relationships and avoiding the oversimplification seen in traditional process mining. This results in more accurate and insightful process models that reflect real-world dynamics.

To explain the challenges of data extraction better, let us make the assumption that each event refers to:

- one activity (e.g., place order)
- one timestamp (e.g., April 5, 2023 6:19 PM)
- any number of objects possibly of different types (e.g., one customer, one order, three items, and one account manager)
- any number of event attributes (e.g., costs).

By dropping the requirement of focusing on a single object (i.e., case), we are able to describe reality better.

Let's make the following assumption

Activity	Time	Orders	Items	Packages	Customers	Products	Price	Weight
pick item	2019/12/26 12:04:46	(991224)	(884803)	()	(Wil van der Aalst)	(iPhone 8)	529.0	0.21
reorder item	2019/12/26 12:27:26	(991271)	(885002)	()	(Mohammadreza Fori Sani)	(Kindle Paperwhite)	129.0	0.496
place order	2019/12/26 12:44:23	(991283)	(885038, 885039)	()	(Luis Santos)	(MacBook Air, iPad)	2700.0	1.733
pick item	2019/12/26 14:01:16	(991268)	(884933)	()	(Naruse Ryosuke)	(MacBook Air)	2200.0	1.26
create package	2019/12/26 14:01:16	(991264)	(884976, 884974, 884978, 884971, 884970, 884973)	(880798)	(Seran Uysal)	(Fire Stick 4K, iPad Pro, iPad Pro, iPad Pro, Fire Stick, Kindle)	3506.97	2.412
send package	2019/12/26 14:16:31	(991264)	(884976, 884974, 884978, 884971, 884970, 884973)	(880798)	(Seran Uysal)	(Fire Stick 4K, iPad Pro, iPad Pro, iPad Pro, Fire Stick, Kindle)	3506.97	2.412
pick item	2019/12/26 14:16:48	(991279)	(885027)	()	(Claudia Graf)	(iPhone 11)	799.0	0.166
confirm order	2019/12/26 14:26:01	(991283)	(885038, 885039)	()	(Luis Santos)	(MacBook Air, iPad)	2700.0	1.733
reorder item	2019/12/26 14:32:43	(991251)	(884912)	()	(Tobias Brockhoff)	(Fire Stick)	39.99	0.2
confirm order	2019/12/26 14:32:44	(991272)	(885038, 885037)	()	(Seran Uysal)	(Kindle Paperwhite, iPad Air, iPad Pro)	134.98	1.26
pick item	2019/12/26 14:33:28	(885024)	(885024)			(MacBook Air)	2200.0	1.26
place order	2019/12/26 14:48:33	(991284)	(885040, 885041, 885042)			(one, Fire Stick, MacBook Air, Echo Show 8, Fire TV Pro)	4222.98	2.79
failed delivery	2019/12/26 15:04:53	(991240, 991168)	(884879, 884961, 884873)			(Kindle Echo Studio, Echo Studio, Kindle, Mia Echo, iPad mini, iPad Pro, iPad Pro)	5982.95	7.642
pick item	2019/12/26 15:20:05	(991278)	(885026)			(one 8)	699.0	0.172
confirm order	2019/12/26 15:25:00	(991245)	(884938, 884939, 884940)			(Kindle Paperwhite)	3247.95	2.066
send package	2019/12/26 15:25:42	(991249, 881253)	(884902, 884922, 884921)			(MacBook Air, iPad)	2406.0	4.054
failed delivery	2019/12/26 15:35:16	(991264)	(884976, 884974, 884978)			(MacBook Air)	3506.97	2.412
confirm order	2019/12/26 15:40:51	(991274)	(885008, 885009, 885010)			(Kindle, iPhone X, Fire Stick, iPhone 8)	1352.98	1.065
failed delivery	2019/12/26 15:46:21	(99128, 991255)	(884426, 884932, 884999)			(Echo Show 8, Kindle Paperwhite, iPad mini, Kindle, iPhone X, iPhone 8, Echo Show)	2145.97	3.6
payment reminder	2019/12/26 15:54:44	(991169)	(884566, 884566, 884567, 884568)		(Gyuram Park)	(iPhone 8, Echo Plus, iPad Air, iPad mini)	1808.99	2.21
pick item	2019/12/26 15:55:38	(991201)	(884717)		(Seran Uysal)	(Echo Show 8)	129.99	0.98
pick item	2019/12/26 16:00:38	(991251)	(884912)		(Tobias Brockhoff)	(Fire Stick)	39.99	0.2
reorder item	2019/12/26 16:04:42	(991265)	(884977)		(Seran Uysal)	(Fire Stick 4K)	89.99	0.28
payment reminder	2019/12/26 16:11:39	(991164)	(884542, 884543, 884544, 884545, 884546, 884547)		(Junkang Gao)	(Kindle Paperwhite, iPad Air, iPhone 11, MacBook Air, iPad mini, Echo Dot)	4087.99	3.01
pick item	2019/12/26 16:22:04	(991241)	(884882)		(Lisa Manne)	(iPhone 8)	529.0	0.21
create package	2019/12/26 16:22:04	(991263, 991162)	(884967, 884964, 884966)	(880799)	(Luis Santos)	(iPad Air, iPhone 8, iPad)	1500.0	1.131

event = activity + timestamp + objects + attributes

This assumption leads to example events that have varying numbers of objects involved.



# Let's make the following assumption

Activity	Time	Orders	Items	Packages	Customers	Products	Price	Weight
pick item	2019/12/26 12:04:46	{991224}	{884863}	{}	{Williama cher-Ardet}	{iPhone 8}	529.0	0.21
reorder item	2019/12/26 12:27:26	{991271}	{885002}	{}	{}	{}	123.0	0.496
place order	2019/12/26 12:42:23	{991263}	{885038, 885039}	{}	{}	{}	2200.0	1.753
pick item	2019/12/26 14:01:36	{991266}	{884983}	{}	{}	{}	2200.0	1.26
create package	2019/12/26 14:01:36	{991264}	{884975, 884974, 884976, 884971, 884970, 884973}	{660796}	{}	{}	3506.97	2.412
send package	2019/12/26 14:16:11	{991264}	{884975, 884974, 884976, 884971, 884970, 884973}	{660796}	{}	{}	3506.97	2.412
pick item	2019/12/26 14:16:48	{991279}	{885027}	{}	{Claudia Graf}	{iPhone 11}	799.0	0.166
confirm order	2019/12/26 14:26:01	{991263}	{885038, 885039}	{}	{Luis Santos}	{MacBook Air, iPad}	2200.0	1.753
reorder item	2019/12/26 14:32:43	{991261}	{884922}	{}	{Tobias Brockhoff}	{Fire Stick}	39.99	0.2
confirm order	2019/12/26 14:32:44	{991272}	{885036, 885037}	{}	{Lisa Manne}	{Echo, Echo Dot}	134.98	1.16
pick item	2019/12/26 14:33:28	{885024}	{885024}	{}	{Junxiong Gao}	{MacBook Pro}	2500.0	1.37
failed delivery	2019/12/26 15:40:51	{991274}	{885006, 885007, 885008, 885009}	{}	{Christine Dabbert}	{iPhone, Fire Stick, MacBook Air, Echo Show 8, iPhone 11 Pro}	4222.98	2.79
			{884926, 884932, 884939, 885008, 885009, 885011, 884903}	{}	{}	{}		
failed delivery	2019/12/26 15:46:21	{991285, 991286}	{884973, 884913, 884976, 884938, 884914, 884941, ...}	{660790}	{Tobias Brockhoff}	{iPad Air, Echo Studio, Echo Studio, Kindle, Kindle Echo, iPad mini, iPad Pro, iPad Pro}	5982.95	7.642
			{884940, 884941, 884942, 884943}	{}	{Junxiong Gao}	{iPhone 8}	529.0	0.172
confirm order	2019/12/26 15:46:21	{991274}	{885006, 885007, 885008, 885009}	{}	{Tobias Brockhoff}	{}		
			{884923, 885004, 885005, 884901}	{660796}	{Mohammadreza Fani Sani}	{}		
failed delivery	2019/12/26 15:46:21	{991285, 991286}	{884973, 884976, 884970, 884973}	{660796}	{Seran Uyulal}	{}		
			{884923, 885004, 885005, 884901}	{660796}	{Junxiong Gao}	{}		
payment reminder	2019/12/26 15:54:44	{991169}	{884565, 884566, 884567, 884568}	{}	{Gyungnam Park}	{}		
pick item	2019/12/26 15:55:38	{991201}	{884719}	{}	{Seran Uyulal}	{}		
pick item	2019/12/26 15:55:38	{991265}	{884977}	{}	{Tobias Brockhoff}	{}		
reorder item	2019/12/26 16:04:42	{991265}	{884977}	{}	{Seran Uyulal}	{}		
payment reminder	2019/12/26 16:11:39	{991164}	{884542, 884543, 884544, 884545, 884546, 884547}	{}	{Junxiong Gao}	{}		
pick item	2019/12/26 16:22:04	{991241}	{884882}	{}	{Lisa Manne}	{iPhone 8}	529.0	0.21
create package	2019/12/26 16:22:04	{991263, 99162}	{884967, 884964, 884969}	{660799}	{Luis Santos}	{iPad Air, iPhone 8, iPad}	1500.0	1.133

this "place order" event refer to two items

every "pick item" event refers to one item

this "failed delivery" event refers to one package, seven items, two orders, etc.

event = activity + timestamp + objects + attributes

In the table, there are five object types: orders, items, packages, customers, and products. To create a traditional event log we need to promote one of these object types to the selected case notion and transform the data. If we pick the object type *order* as a case notion, then all events that do not refer to an order disappear, and all events that refer to multiple orders get replicated. After this, we have a traditional event log with one case identifier per event.

## Order as a case notion

activity	time	orders	items	packages
...	...	...	...	...
place order	2020-6-20	{99001}	{88001, 88002}	{}
pick items	2020-6-22	{99001}	{88001}	{}
pick item	2022-6-23	{99001}	{88002}	{}
...	...	...	...	...
send package	2020-6-25	{99001, 99002}	{88002, 88003, 88004}	{66001}
...	...	...	...	...

Events may be duplicated

activity	time	case	items	packages
...	...	...	...	...
place order	2020-6-20	99001	{88001, 88002}	{}
pick item	2022-6-22	99001	{88001}	{}
pick item	2022-6-23	99001	{88002}	{}
...	...	...	...	...
send package	2020-6-25	99001	{88002, 88003, 88004}	{66001}
send package	2020-6-25	99002	{88002, 88003, 88004}	{66001}
...	...	...	...	...

If we pick the object type *item* as a case notion, then all events that do not refer to an item disappear, and all events that refer to multiple items get replicated. Again, we get a traditional event log but now with items as cases.

## Item as a case notion

activity	time	orders	items	packages
...	...	...	...	...
place order	2020-6-20	{99001}	{88001, 88002}	{}
pick items	2020-6-22	{99001}	{88001}	{}
pick item	2022-6-23	{99001}	{88002}	{}
...	...	...	...	...
send package	2020-6-25	{99001, 99002}	{88002, 88003, 88004}	{66001}
...	...	...	...	...

Events may be duplicated

activity	time	orders	case	packages
...	...	...	...	...
place order	2020-6-20	{99001}	88001	{}
place order	2020-6-20	{99001}	88002	{}
pick item	2022-6-22	{99001}	88001	{}
pick item	2022-6-23	{99001}	88002	{}
...	...	...	...	...
send package	2020-6-25	{99001, 99002}	88002	{66001}
send package	2020-6-25	{99001, 99002}	88003	{66001}
send package	2020-6-25	{99001, 99002}	88004	{66001}
...	...	...	...	...

If we pick the object type *package* as a case notion, then all events that do not refer to a package disappear, and all events that refer to multiple packages get replicated.

## Package as a case notion

activity	time	orders	items	packages
...	...	...	...	...
place order	2020-6-20	{99001}	{88001, 88002}	{}
pick items	2020-6-22	{99001}	{88001}	{}
pick item	2022-6-23	{99001}	{88002}	{}
...	...	...	...	...
send package	2020-6-25	{99001, 99002}	{88002, 88003, 88004}	{66001}
...	...	...	...	...

Events may disappear

case				
activity	time	orders	items	packages
...	...	...	...	...
send package	2020-6-25	99002	88002	66001
...	...	...	...	...

The examples show that it is possible to convert object-centric event data into traditional event logs. However, the resulting event logs and corresponding models heavily depend on the selected case notion. Each event log can be seen as a viewpoint or a two-dimensional picture of a three-dimensional reality.

### Flattening Event Data Problems

- **Deficiency:** Events in the original event log with no corresponding events in the flattened event log may disappear unintentionally.
- **Convergence:**
  - Events referring to multiple objects of the selected type are replicated, causing unintentional duplication.
  - Example: If an order has multiple items and "item" is chosen as the case identifier, the "place order" event is replicated for each item, leading to incorrect frequency and performance measures.
- **Divergence:**
  - Events referring to different objects of a type not selected as the case notion may appear causally related but are not.
  - Example: Two events involving the same item in an order context are indistinguishable from two events involving different items if "order" is the case notion, losing the distinction between specific items.

### Illustration with a Pit Stop Example

- **Pit Stop:**
  - A high-level activity involving multiple objects (car, driver, pit crew, tires, etc.).
  - Traditional process mining for a single object (e.g., tire) loses the multi-object perspective.
- **Convergence Problem:**
  - Using "tire" as the case notion, the pit stop event appears to happen eight times (once for each tire), leading to replicated data and distorted statistics.
- **Divergence Problem:**
  - Using "car" as the case notion for low-level events (stop, remove tire, mount tire, drive) results in the inability to distinguish between specific tire removals and mountings.
  - Example: Four "remove tire" and four "mount tire" events per pit stop refer to the same case (car), failing to capture the correct sequence (e.g., front left tire removal before mounting).

### Impact on Process Models

- **Misleading Statistics:**
  - Convergence and divergence problems lead to incorrect interpretations of frequencies, times, and deviations.
- **Overly Complex Models:**
  - Simplified process models (e.g., Directly Follows Graphs) fail to capture specific relationships, leading to "spaghetti-like" models that obscure clear patterns.

### Object-Centric Process Mining (OCPM)

- **Advantages:**
  - Maintains relationships between objects and events, providing a more accurate view.
  - Addresses convergence and divergence issues by using representations closer to reality.



## Real-World Relevance

- **Sales Orders and Items:**
  - Similar phenomena as in the pit stop example.
  - Events for sales order items follow clear patterns (e.g., produce before ship) which are lost when using sales orders as the case notion.
  - Leads to complex and misleading process models.

## Conclusion

- **OCPM:**
  - Offers a more nuanced approach to process mining by avoiding the pitfalls of traditional methods.
  - Enhances the accuracy of process models and statistics by considering multiple objects and their relationships.

## ▼ Exercise

### Question 1

1/1 point (ungraded)

Select the correct statements.

☒ In object-centric event logs, an event can relate to a collection of objects.

☒ In object-centric event logs, an event does not necessarily relate to a single case identifier.

☒ The choice of case notion has a huge impact on the discovered process models.



## Question 2

1/1 point (ungraded)

event_id	timestamp	activity	weight	items	orders
206	5/24/2019 15:54	package delivered	4.273	['880101', '880098', '880003']	['990025', '990001']
207	5/24/2019 15:56	reorder item	0.2	['880102']	['990025']
208	5/24/2019 15:56	reorder item	0.28	['880108']	['990026']
209	5/24/2019 16:25	place order	2.969	['880140', '880142', '880143', '880144']	['990036']
210	5/24/2019 16:32	confirm order	2.583	['880106', '880107', '880105']	['990026']

Consider the object centric event log above.

To create a traditional event log, how often do we duplicate the event with event-id 209 if "items" is selected as the case notion?



**Answer**

Correct: There are four items involved in the event.

How often do we duplicate the event with event-id 206 if "orders" is selected as the case notion?



**Answer**

Correct: There are two orders involved in the event.

## Question 3

1/1 point (ungraded)

1. Which problem does the following sentence describe?

"Events referring to multiple objects of the selected type are replicated, possibly leading to unintentional duplication."

Convergence



2. Which problem does the following sentence describe?

"Events in the original event log with no corresponding events in the flattened event log disappear from the data set."

Deficiency



3. Which problem does the following sentence describe?

"Events referring to different objects of a type not selected as the case notion are considered to be causally related"

Divergence



## ▼ Reshaping Process Mining Using Object-Centric Event Data

*Object-Centric Event Data* (OCED) allow events to point to any number of objects rather than a single case. *OCEL 2.0* (<https://ocel-standard.org/>) is a standard format for storing OCED. By converting OCED to a conventional event log, we can apply all existing process mining techniques, but this may lead to the convergence and divergence problems described before. However, it is possible to extend existing process mining techniques to deal with OCED natively. OCED data can be used to discover object-centric process models (e.g., object-centric Petri nets, as shown in <https://doi.org/10.3233/FI-2020-1946>). These discovered object-centric process models are annotated with correct event and object frequencies, thus avoiding incorrect diagnostic information. Problems related to the unintentional duplication of events (convergence) and the loss of causal information (divergence) can be avoided. Moreover, object-centric process models can also be used for conformance checking.

Object-Centric Process Mining (OCPM) provides the following advantages:

- *Data configuration is only done once.* After the data is loaded, the user chooses the view of objects and events that they want to look at. There is no need to extract and transform the data repeatedly for questions requiring a different viewpoint.
- *Interactions between objects are captured.* The root causes of many performance and compliance problems can only be explained by considering the interactions between different objects.
- *3-dimensional process mining data models to better represent reality.* OCPM creates models that are 3-dimensional and easily extensible to add more objects and events to capture more use cases for process mining.

A detailed treatment of OCPM is out of scope. However, in the future, this topic will become more important. In 2022, Celonis released ProcessSphere, supporting object-centric process discovery and performance analysis. This followed research prototypes like [OCPM](#) and [OCpi](#) and the first [OCEL standard](#).

Existing process mining capabilities, ranging from process discovery and conformance checking to predictive analytics and automated actions, can benefit from OCPM. This means that all preexisting types of process mining can be lifted from 2D to 3D, thus revolutionizing the entire industry.

For more information, we refer to the following three papers:

1. W. van der Aalst. Object-Centric Process Mining: Unraveling the Fabric of Real Processes. *Mathematics* 2023, 11, 2691. <https://doi.org/10.3390/math11122691>.
2. W. van der Aalst. Object-Centric Process Mining: The Next Frontier in Business Performance (Whitepaper), March 2023, <https://celonis.com/OCPM-Whitepaper>.
3. W. van der Aalst and A. Berti. Discovering Object-Centric Petri Nets. *Fundamenta Informaticae*, 175(1-4):1-40, 2020, <https://doi.org/10.3233/FI-2020-1946>.

## ▼ Celonis - Getting The Data

Get the data from source systems and which practical challenges occur on the way. We will revisit the concept of an event log and talk about the practical challenges of composing an event log in real life.

In a second step

will walk you through a CSV data upload in Celonis and create a data model containing several tables and key them together with you. We will also visit a real-life data model for a P2P case generated in SAP and talk about role of connectors in getting the data.

You can find the data sets for this hands-on session below, together with a tutorial on how to upload data in the new Celonis interface:

- <https://courses.edx.org/asset-v1:RWTHx+PM+1T2023+type@asset+block@P2P-Cases.csv>
- <https://courses.edx.org/asset-v1:RWTHx+PM+1T2023+type@asset+block@P2P-Cases.csv>
- <https://www.youtube.com/watch?v=QvkFpjsB4lc>