

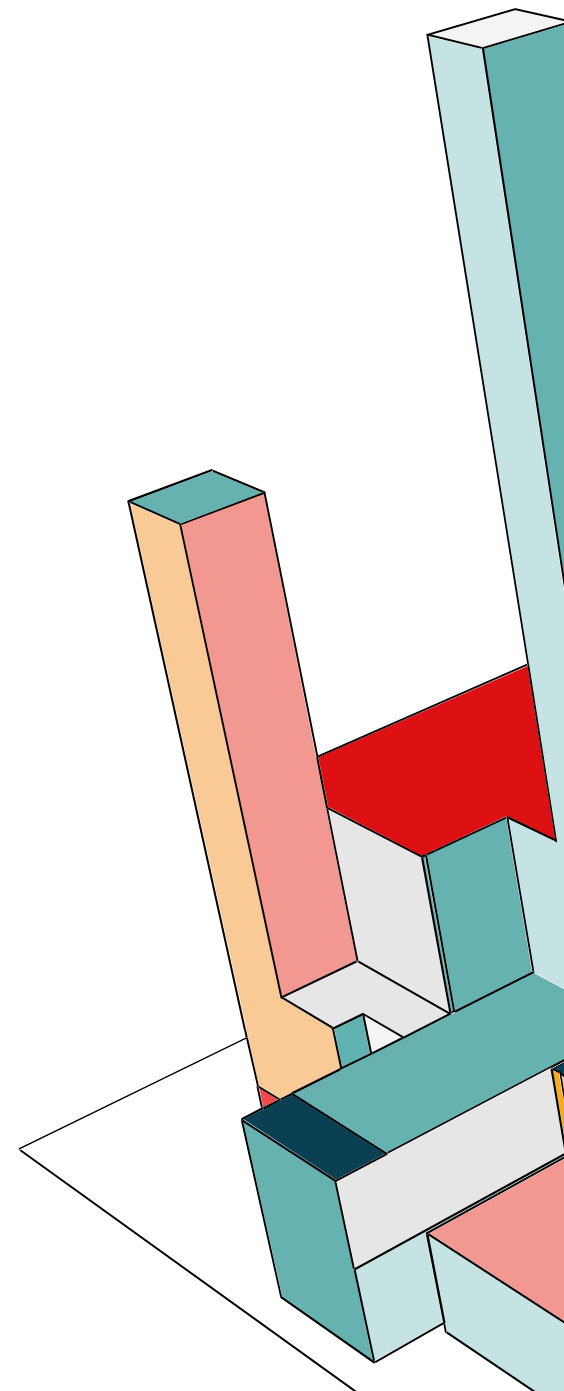
تصميم وهيكلة البرمجيات **ITSE411**

المحاضرة السادسة

Object-oriented design using UML diagrams

محاور المحاضرة

- Activity Diagram
- State-machine Diagrams
- Sequence Diagram
- Communication Diagram



UML ACTIVITY DIAGRAM

- **activity diagram** is used to describe A use case model. However, to depict a use case, a subset of the activity diagram capabilities is sufficient. In particular, it is not necessary to model concurrent activities for use cases.
- An activity diagram can be used to represent the sequential steps of a use case, including the main sequence and all the alternative sequences.
- Activity diagrams is something like the famous flow charts but it's much more powerful than flow charts. Flow charts are not part of UML diagrams.

- يُستخدم مخطط النشاط Activity Diagram لوصف نموذج حالة الاستخدام Use Case Model. ومع ذلك، فإن تمثيل حالة استخدام لا يتطلب الاستفادة من جميع قدرات مخطط النشاط؛ إذ يكفي استخدام جزء منها فقط. على سبيل المثال، ليس من الضروري تمثيل الأنشطة المتزامنة عند التعامل مع حالات الاستخدام.
- يمكن لمخطط النشاط أن يُستخدم لتمثيل الخطوات المتتالية لحالة الاستخدام، بما في ذلك التسلسل الرئيسي وجميع التسلسلات البديلة.
- ويُشبه مخطط النشاط إلى حد ما المخططات الانسيابية Flow Charts الشهيرة، لكنه أكثر قوة وثراءً في التعبير. ومن المهم الإشارة إلى أن المخططات الانسيابية لا تُعد جزءاً من مخططات UML



□ key components and concepts of UML activity diagrams:

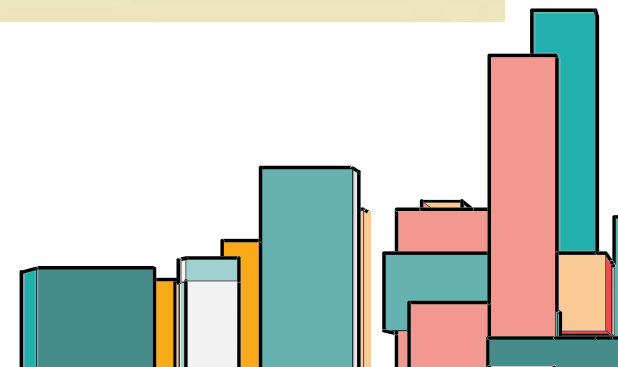
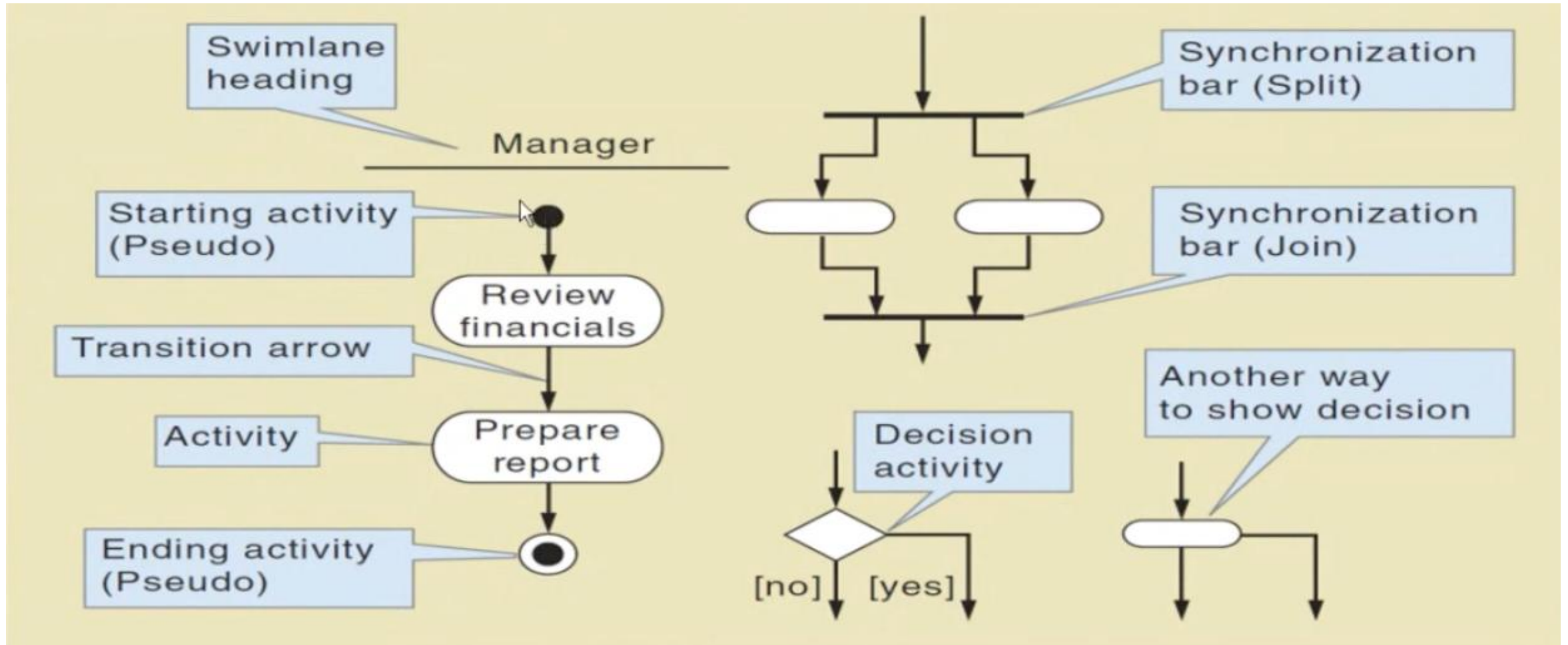
- **Initial and Final Nodes:** Initial nodes represent the start of the activity diagram, while final nodes represent the end.
- **Activities:** Activities represent tasks or actions that occur within the system.
- **Decisions:** Decisions, also known as decision nodes or decision points, represent points in the process where the flow of control can diverge based on conditions.
- **Merge Nodes:** Merge nodes are used to synchronize multiple incoming transitions into a single outgoing transition.
- **Forks and Joins:** Forks and joins are used to create parallel paths in the process flow. A fork splits the flow of control into multiple concurrent paths, while a join merges multiple concurrent paths back into a single path.

UML ACTIVITY DIAGRAM

- **العقدة الابتدائية Initial Node:** تمثل نقطة البداية في مخطط النشاط. **العقدة النهائية Final Node:** تمثل نقطة انتهاء النشاط أو اكتمال سير العملية.
- **الأنشطة تمثل المهام أو الأفعال التي يقوم بها النظام.** كل نشاط يعكس خطوة محددة ضمن سير العمل.
- **تُعرف أيضًا باسم Decision Nodes أو Decision Points.** تمثل نقاط في العملية يمكن أن يتفرع عندها سير التحكم بناءً على شروط أو حالات مختلفة.
- **تُستخدم عقد الدمج** لدمج عدة انتقالات واردة في انتقال واحد خارج. تساعد على إعادة توحيد المسارات المتفرعة في مسار واحد.
- **التفرعات والانضمامات Forks and Joins**
 - **Fork تفرع:** يُستخدم لتقسيم سير التحكم إلى عدة مسارات متزامنة تعمل بالتوازي.
 - **Join انضمام:** يُستخدم لدمج عدة مسارات متزامنة مرة أخرى في مسار واحد.



UML ACTIVITY DIAGRAM



An example of an activity diagram for write prescription use case of the Clinic system

مثال على مخطط النشاط لحالة استخدام كتابة وصفة طبية في نظام العيادة. يُستخدم مخطط النشاط لتمثيل تسلسل الخطوات والإجراءات بين المريض والدكتور بشكل بصري ومنظم، حيث يوضح كيف تبدأ العملية وتنتهي، وما هي الأنشطة التي يقوم بها كل طرف.

في هذا السيناريو:

✓ تبدأ العملية من العقدة الابتدائية Initial Node .

✓ يقوم الدكتور بفحص المريض كخطوة أولى.

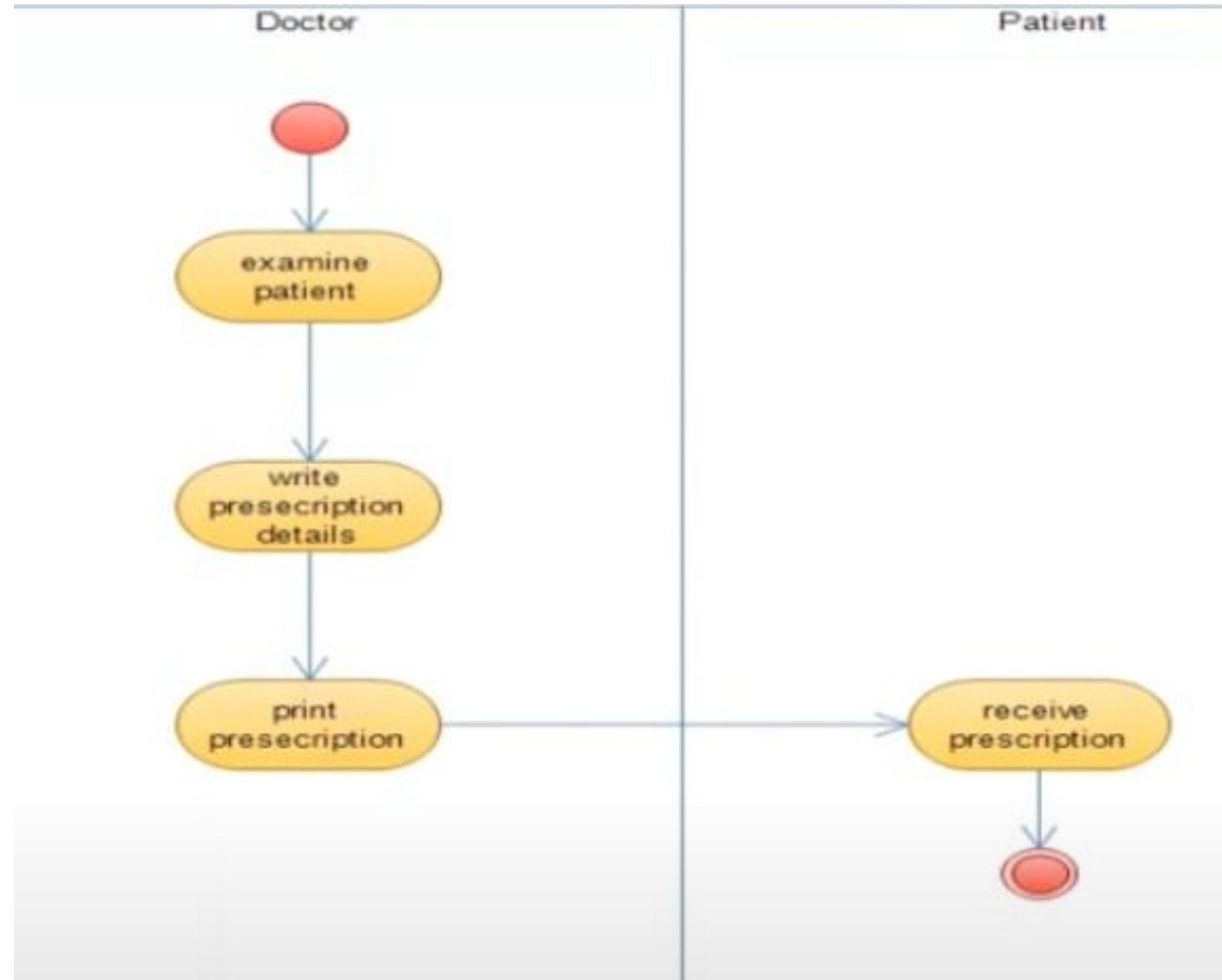
✓ بعد ذلك يقوم الدكتور بكتابة التشخيص والتفاصيل الطبية.

✓ ثم يتم طباعة التشخيص والوصفة الطبية.

✓ أخيرًا، يقوم المريض باستلام العلاج بناءً على الوصفة المطبوعة.

✓ تنتهي العملية عند العقدة النهائية Final Node .

UML ACTIVITY DIAGRAM

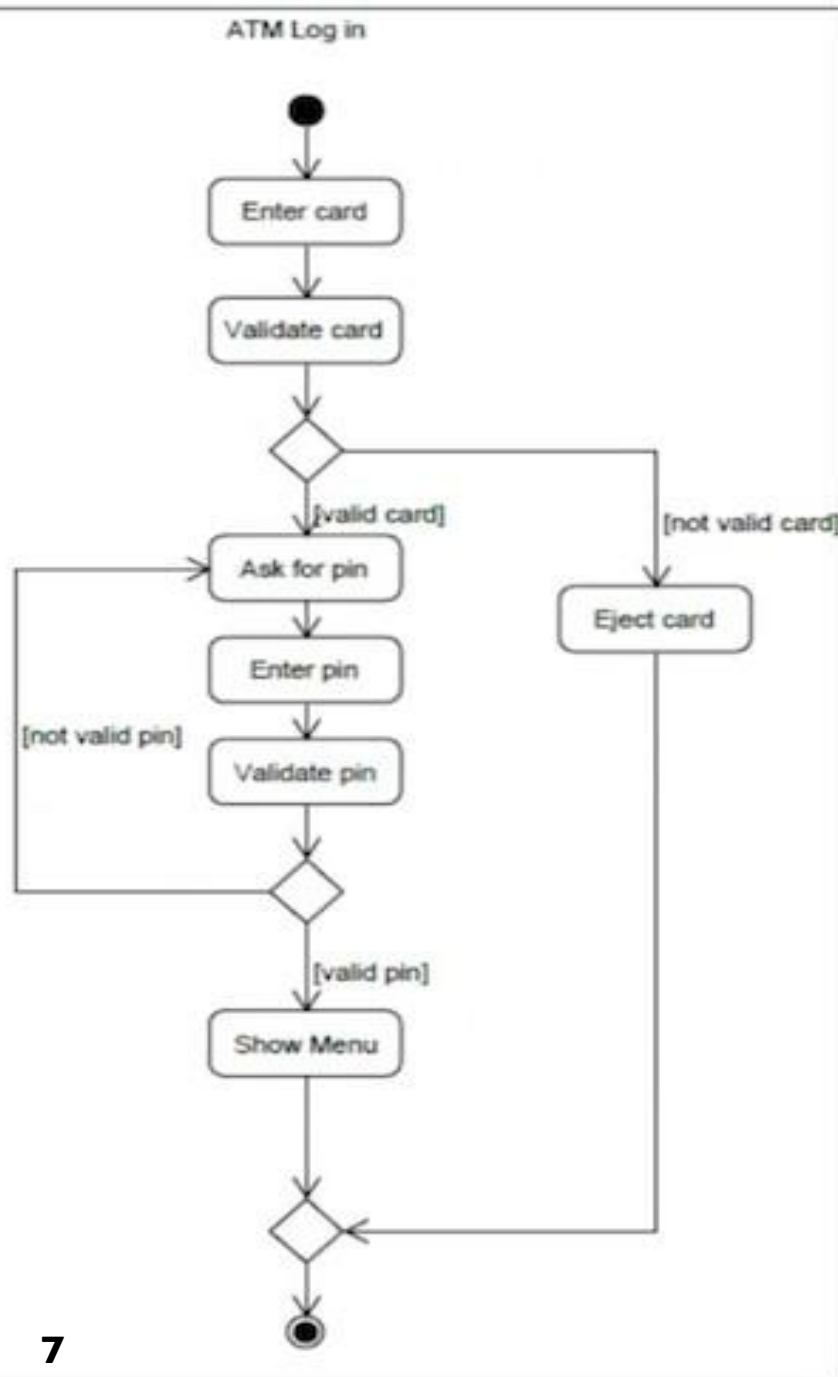


UML ACTIVITY DIAGRAM

مثال: مخطط نشاط لدخول إلى جهاز الصراف الآلي

♦ خطوات الخوارزمية

An example of an activity diagram for write prescription use case of ATM LOGIN



1. Initial Node البداية تبدأ العملية من العقدة الابتدائية.

2. Insert Card إدخال البطاقة يقوم المستخدم بإدخال البطاقة في الجهاز.

3. Decision Node: Card Valid? التحقق من البطاقة

○ إذا كانت البطاقة صالحة → الانتقال إلى الخطوة التالية.

○ إذا كانت البطاقة غير صالحة → رفض البطاقة والانتقال مباشرة إلى العقدة النهائية.

4. Enter PIN إدخال الرقم السري يسأل النظام عن الرقم السري، والمستخدم يقوم بإدخاله.

5. Decision Node PIN Valid? التحقق من الرقم السري

○ إذا كان الرقم السري صحيحًا → الانتقال إلى عرض القائمة.

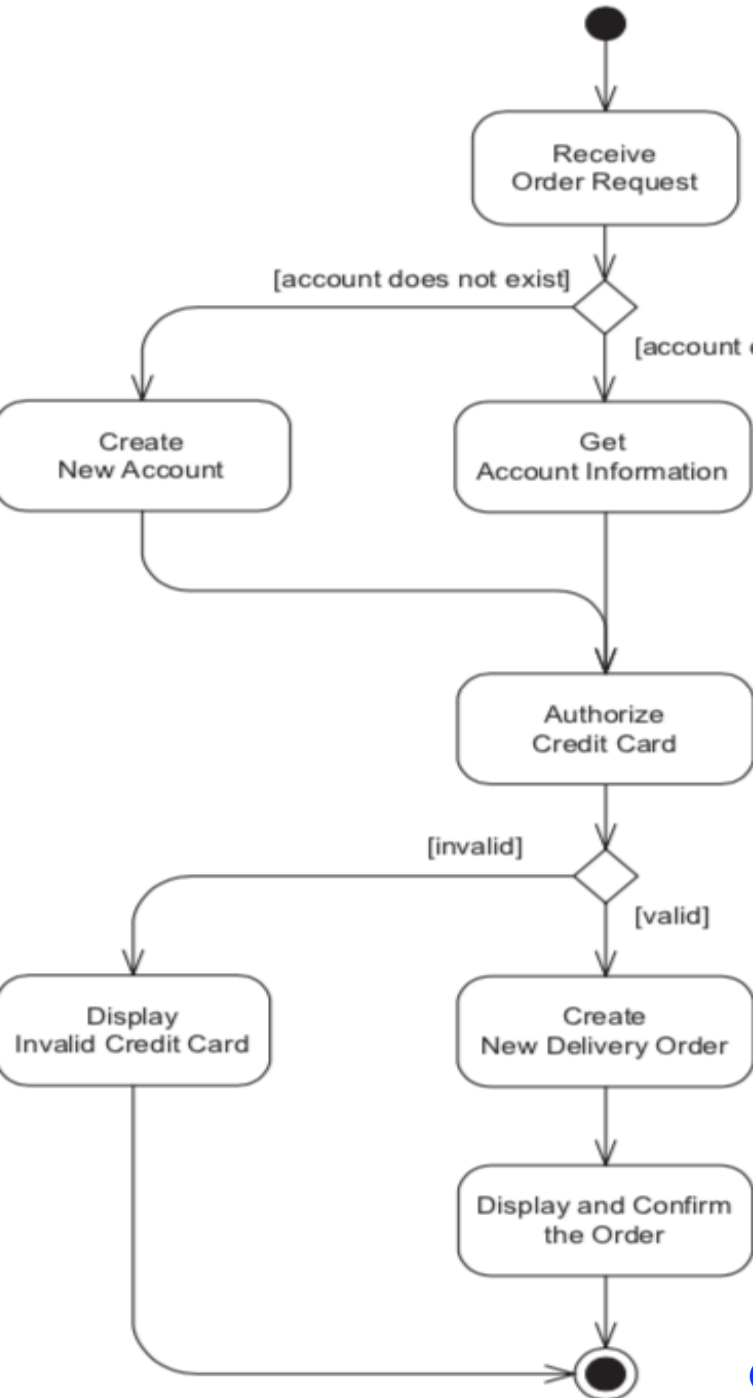
○ إذا كان الرقم السري خاطئًا → العودة مرة أخرى لطلب إدخال الرقم السري.

6. Show Menu عرض القائمة بعد التحقق بنجاح، يظهر النظام قائمة الخدمات المتاحة.

7. Final Node النهاية تنتهي العملية عند العقدة النهائية (دائرة سوداء مزدوجة).

UML ACTIVITY DIAGRAM

An example of an activity diagram for the Make Order Request use case of the Online Shopping System



Order processing

1. **Initial Node** البداية تبدأ العملية من العقدة الابتدائية.

2. **Receive Order Request** استقبال طلب إنشاء طلبية النظام يستقبل طلبية جديدة من المستخدم.

3. **Decision Node: Account Exists?** هل الحساب موجود؟

- إذا كان الحساب موجودًا → متابعة.

- إذا لم يكن موجودًا → الانتقال إلى عملية **Get Account Information** إدخال بيانات الحساب/إنشاء حساب جديد .

4. **Authorize Credit Card** تفويض بطاقة الدفع (النظام يتحقق من صلاحية بطاقة الدفع).

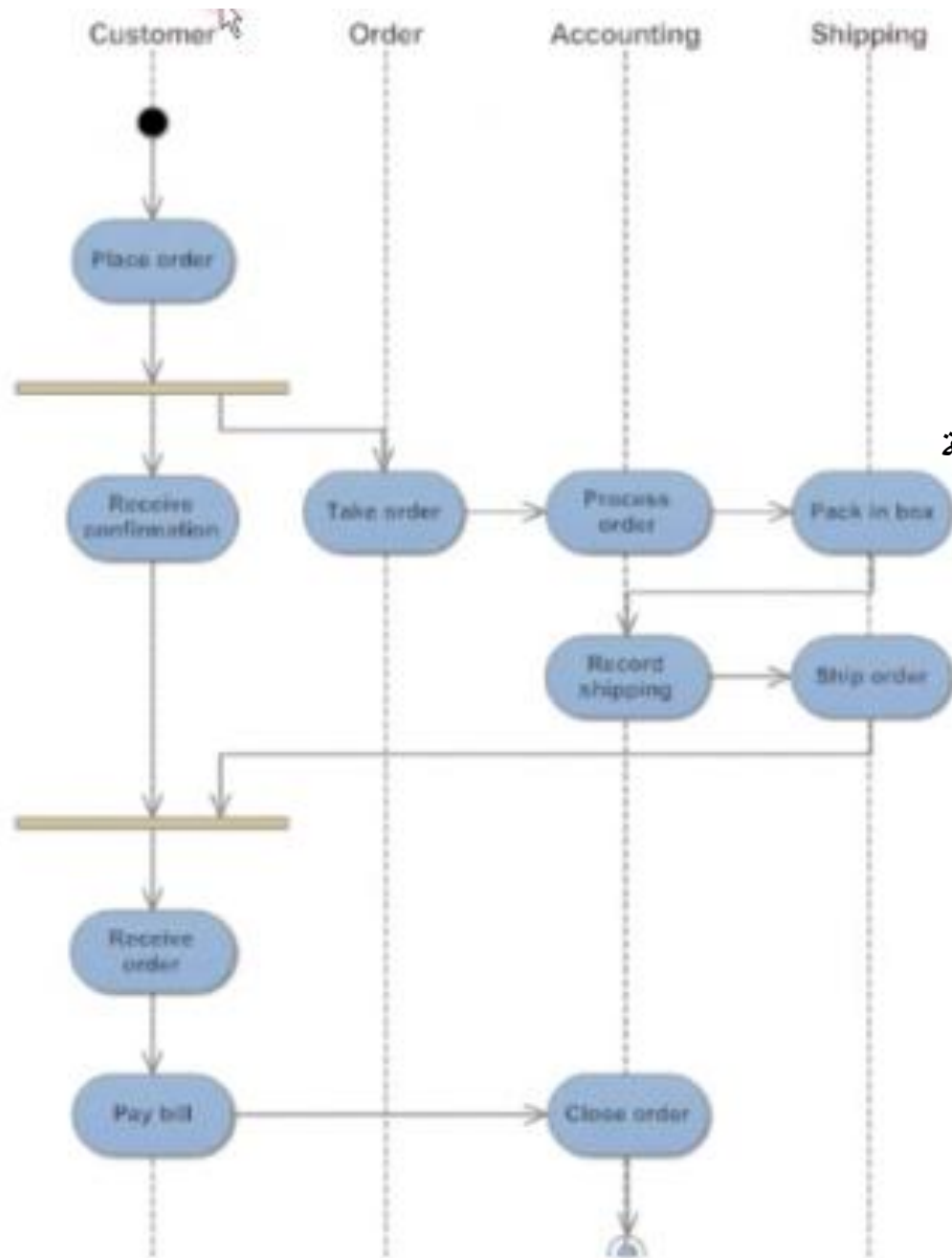
- إذا البطاقة غير صالحة → رفض العملية → النهاية.

- إذا البطاقة صالحة → متابعة.

5. **Create New Delivery Order** إنشاء طلبية توصيل جديدة (النظام يقوم بإنشاء طلبية توصيل بناءً على تفاصيل الطلب والمستخدم).

6. **Display and Confirm the Order** عرض وتأكيـد الطلبية (النظام يعرض تفاصيل الطلبية (المنتجات، الكمية، السعر، عنوان التوصيل)).

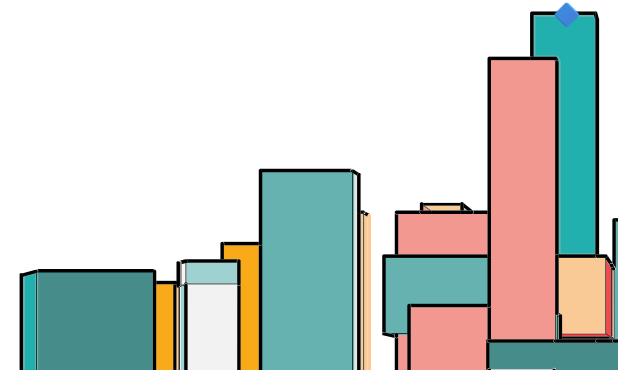




UML ACTIVITY DIAGRAM

SWIMLANE: SWIMLANES GROUP RELATED ACTIVATES INTO ONE COLUMN.

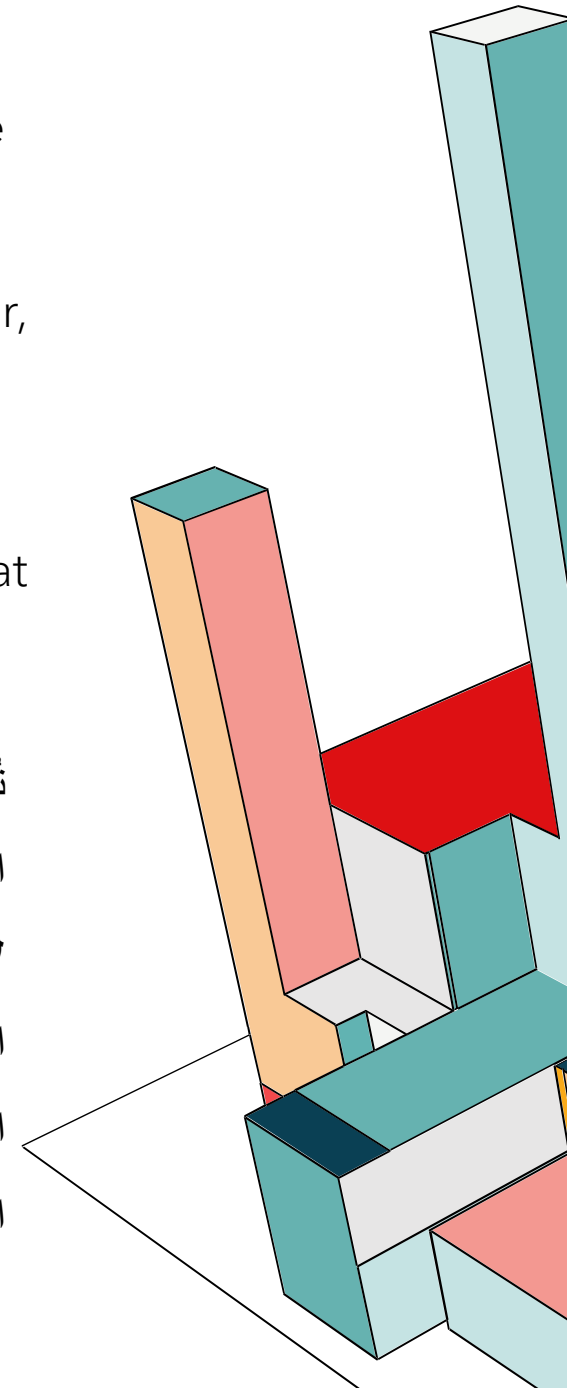
- ♦ Swimlane هو أسلوب لتقسيم مخطط النشاط إلى أعمدة أو صفوف (أشبه بمسارات السباحة في المسبح). كل Swimlane يمثل جهة أو فاعل أو وحدة مسؤولة عن مجموعة من الأنشطة. الهدف هو توضيح من يقوم بأي نشاط داخل سير العمل.
- ♦ يساعد على تنظيم الأنشطة في المخطط بشكل أوضح. يوضح المسؤوليات بين مختلف الجهات أو الكيانات. ويسهل قراءة المخطط وفهمه، خصوصاً في الأنظمة المعقدة.
- ♦ يتم تقسيم المخطط إلى أعمدة أو صفوف. كل عمود/صف يمثل جهة معينة (مثل: زبون، طلبية، محاسبة، توصيل). الأنشطة المرتبطة بهذه الجهة تُرسم داخل الـ Swimlane الخاص بها.



UML State Machine Diagrams

- State machine diagrams depict the dynamic behavior of an entity based on its response to events, showing how the entity reacts to various events based on its current state.
- Create a UML state machine diagram to explore the complex behavior of a class, actor, subsystem, or component.
- State Machine Diagrams, also known as state diagrams or statecharts.
- They are particularly useful for modeling the dynamic behavior of objects or systems that have a finite number of states.

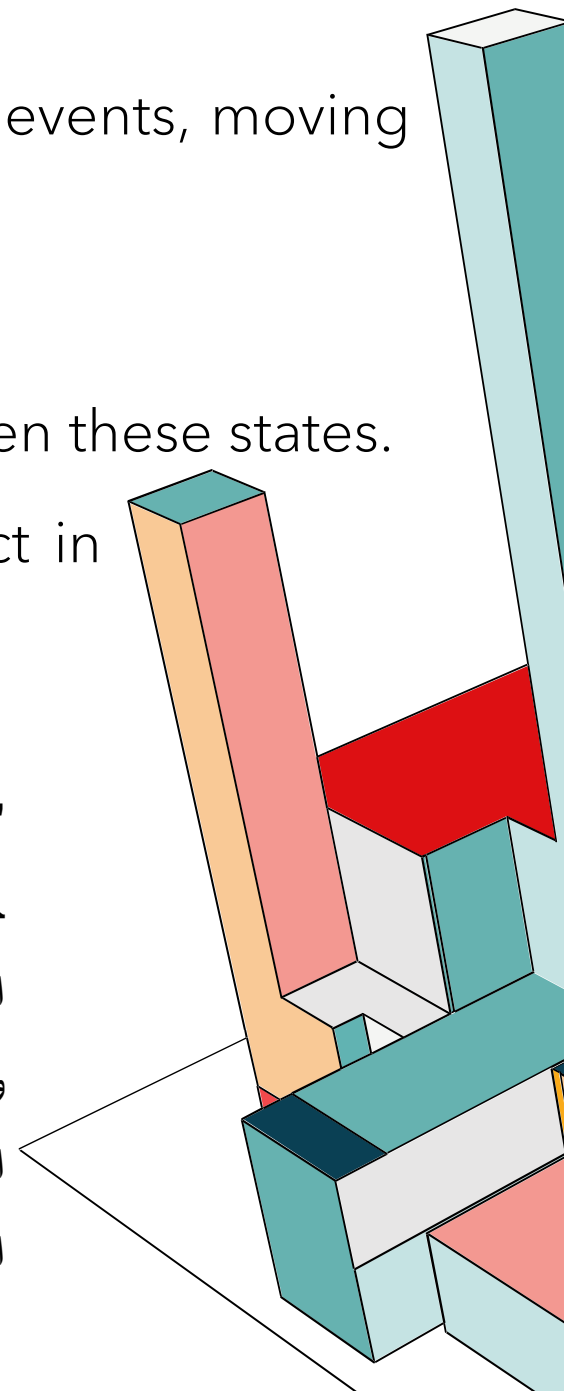
تُستخدم مخططات الحالة State Machine Diagrams لتمثيل السلوك الديناميكي لكيان معين بناءً على استجابته للأحداث المختلفة، حيث تُظهر كيفية تفاعل هذا الكيان مع مجموعة من الأحداث وفقًا لحالته الحالية. تُعد هذه المخططات أداة فعالة لاستكشاف السلوك المعقد للفئات Classes ، أو الجهات الفاعلة Actors ، أو الأنظمة الفرعية Subsystems ، أو المكونات Components ضمن بيئة نمذجة موحدة UML. ويُشار إلى مخططات الحالة أيضًا بمصطلحات مثل "مخططات الحالة" أو "مخططات الحالة البيانية" Statecharts ، وهي مفيدة بشكل خاص في نمذجة السلوك الديناميكي للكائنات أو الأنظمة التي تمتلك عددًا محدودًا من الحالات المحددة.



UML State Machine Diagrams

- State machine diagrams can also show how an entity reacts to various events, moving from one state to another.
- Each class has objects that may have status conditions or "states".
- Object behavior consists of the various states and the movement between these states.
- State Machine Diagram is a diagram which shows the life of an object in states and transitions

"تُظهر مخططات الحالة State Machine Diagrams كيفية استجابة الكيان لمجموعة متنوعة من الأحداث، وذلك من خلال انتقاله بين حالات مختلفة وفقًا لتغير الظروف المحيطة. تتضمن كل فئة Class مجموعة من الكائنات Objects التي قد تمر بحالات أو أوضاع معينة تُعرف باسم "الحالات" States ، والتي تعكس الوضع الحالي للكائن في لحظة معينة. ويُعبر سلوك الكائن عن تسلسله في هذه الحالات، بالإضافة إلى الانتقالات التي تحدث بينها نتيجة لتأثير الأحداث الخارجية أو الداخلية. يُعد مخطط الحالة أداة تصويرية تُستخدم لتمثيل دورة حياة الكائن من خلال الحالات التي يمر بها والانتقالات المرتبطة بها، مما يتيح فهمًا دقيقًا للسلوك الديناميكي للكائنات داخل النظام."



UML State Machine Diagrams

■ key components and concepts of UML State Machine Diagrams:

- **State**: a condition during an object's life when it satisfies some criterion, performs an action, or waits for an event.
- **Actions**: some activity that must be completed as part of a transition.
- **Transitions**: the movement of an object from one state to another
- **Origin state**: the original state of an object before it begins a transition



○ الحالة State تمثل وضعًا معينًا في دورة حياة الكائن، حيث يحقق فيه شرطًا محددًا، أو يؤدي إجراء معينًا، أو ينتظر وقوع حدث ما.

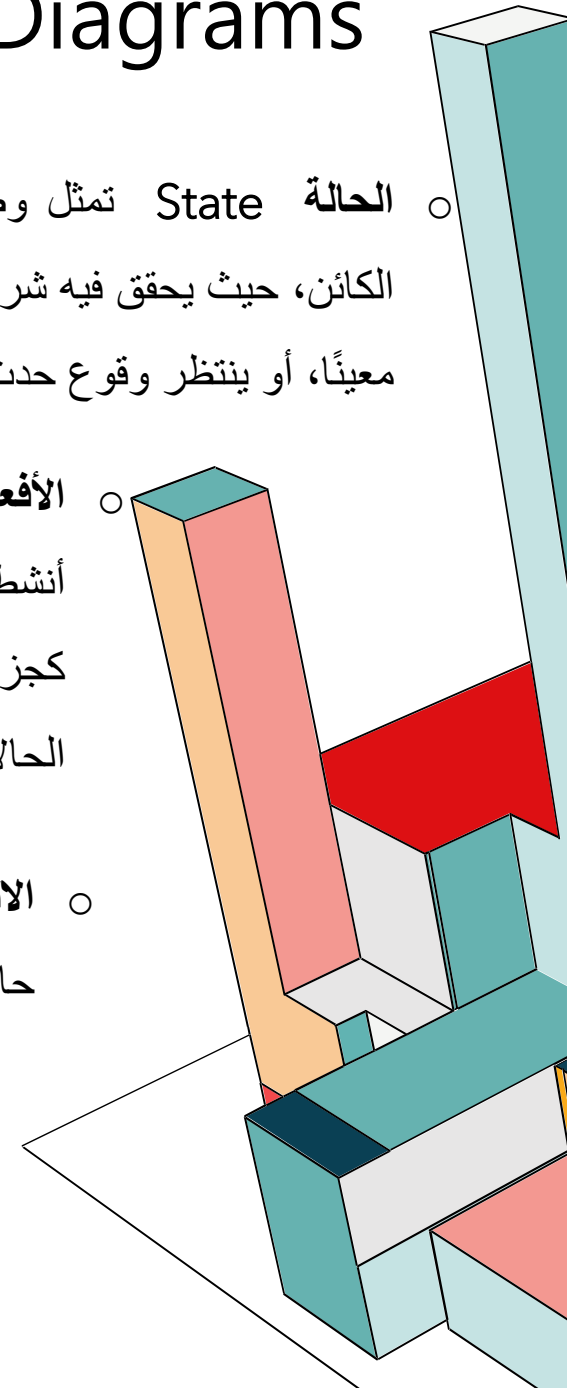


○ الأفعال Actions هي أنشطة أو مهام يجب إنجازها كجزء من عملية الانتقال بين الحالات.

○ الانتقالات Transitions: تشير إلى حركة الكائن من حالة إلى أخرى نتيجة لحدوث حدث معين.

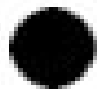


○ الحالة الأصلية Origin State هي الحالة التي يكون عليها الكائن قبل بدء عملية الانتقال.

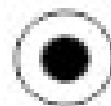


■ key components and concepts of UML State Machine Diagrams:

- Destination state: the state to which an object moves after completing a transition
- guard-condition: a true/false test to see whether a transition can fire
- Initial and Final States
- Pseudostate: the starting point in a state machine diagram. Noted by a black circle.



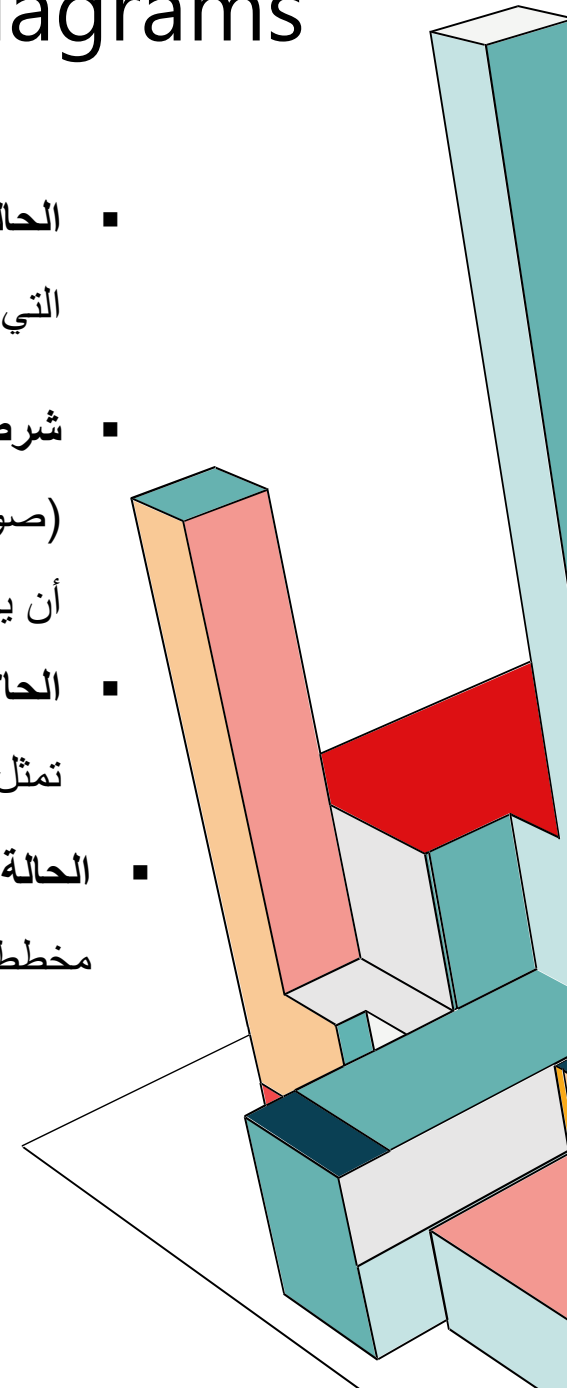
Initial state



Final state

UML State Machine Diagrams

- الحالة الوجهة Destination State هي الحالة التي ينتقل إليها الكائن بعد إتمام عملية الانتقال.
- شرط الحماية Guard Condition اختبار منطقي (صواب/خطأ) يُستخدم لتحديد ما إذا كان الانتقال يمكن أن يحدث أم لا.
- الحالات الابتدائية والنهائية Initial and Final States تمثل نقطة البداية لدورة حياة الكائن ونهايتها ضمن المخطط.
- الحالة الكاذبة Pseudostate تُستخدم لتمثيل نقطة البداية في مخطط الحالة، وغالبًا ما يُرمز إليها بدائرة سوداء.



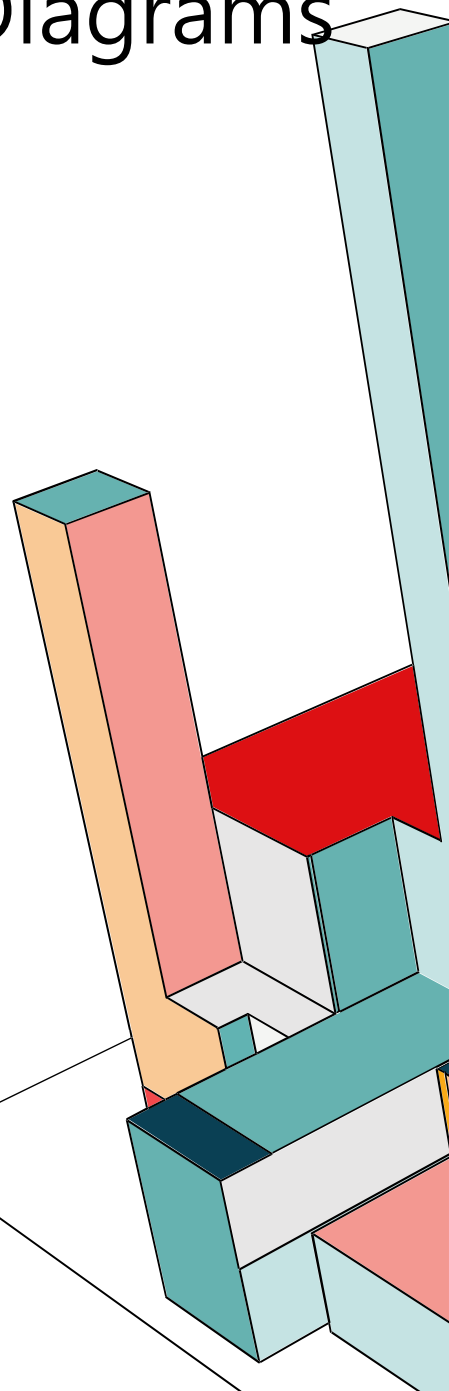
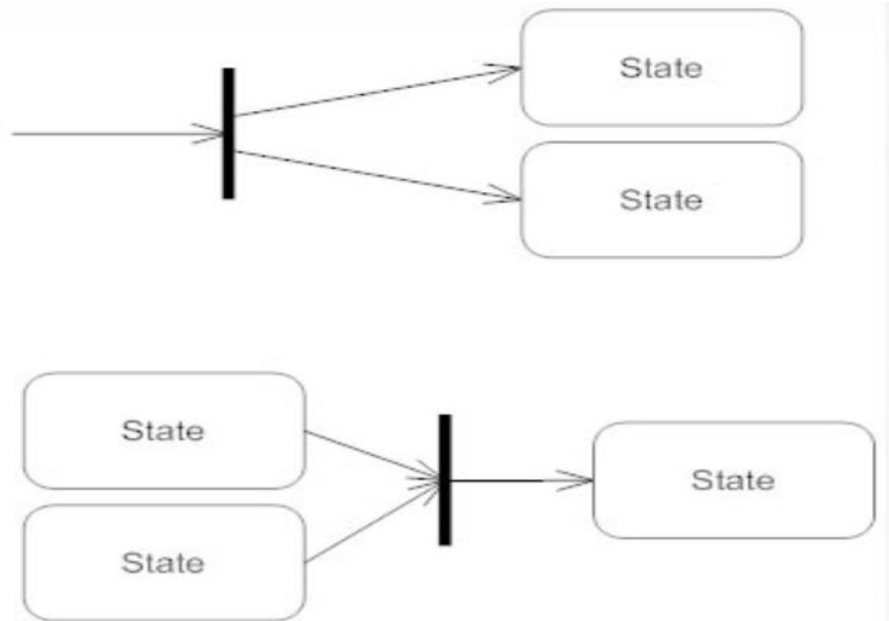
UML State Machine Diagrams

- **Concurrent states:** when an object is in one or more states at the same time.
- **Concurrent paths:** when multiple paths are being followed concurrently, i.e.

when one or more states in one path are

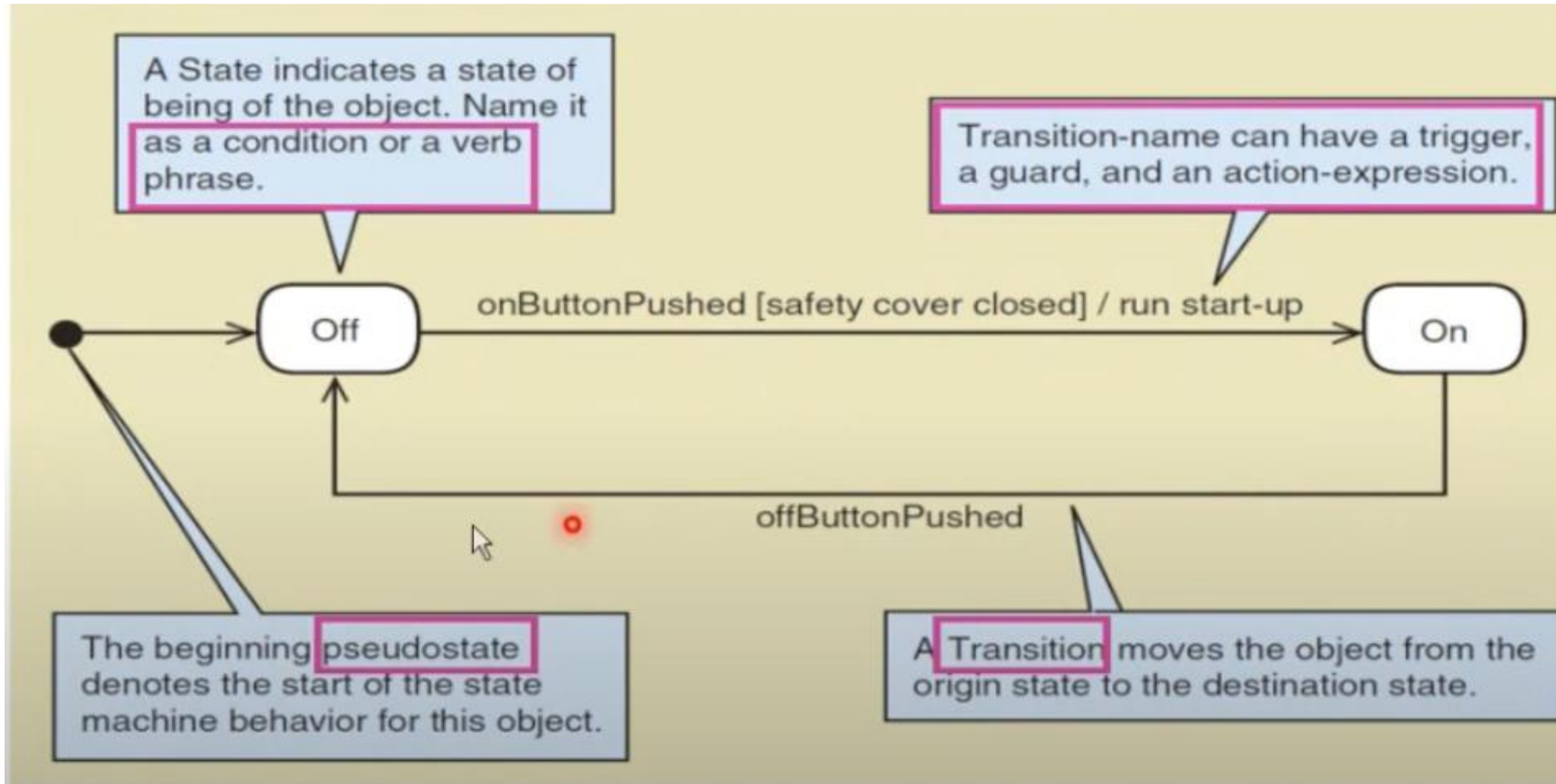
- **الحالات المتزامنة Concurrent States** تشير إلى الوضع الذي يكون فيه الكائن في أكثر من حالة في الوقت نفسه. أي أن دورة حياة الكائن يمكن أن تنقسم إلى حالات متعددة تعمل بشكل متوازي، بحيث يُنفَّذ أكثر من نشاط أو سلوك في آن واحد.

- **المسارات المتزامنة Concurrent Paths** تمثل الحالات أو المسارات التي تُتبع بشكل متزامن داخل المخطط، حيث يمكن أن تكون هناك مجموعة من الحالات في مسار معين تعمل بالتوازي مع حالات أخرى في مسار مختلف. هذا المفهوم يُستخدم لتوضيح السلوكيات المعقدة للكائنات أو الأنظمة التي تتطلب معالجة متعددة في نفس اللحظة.



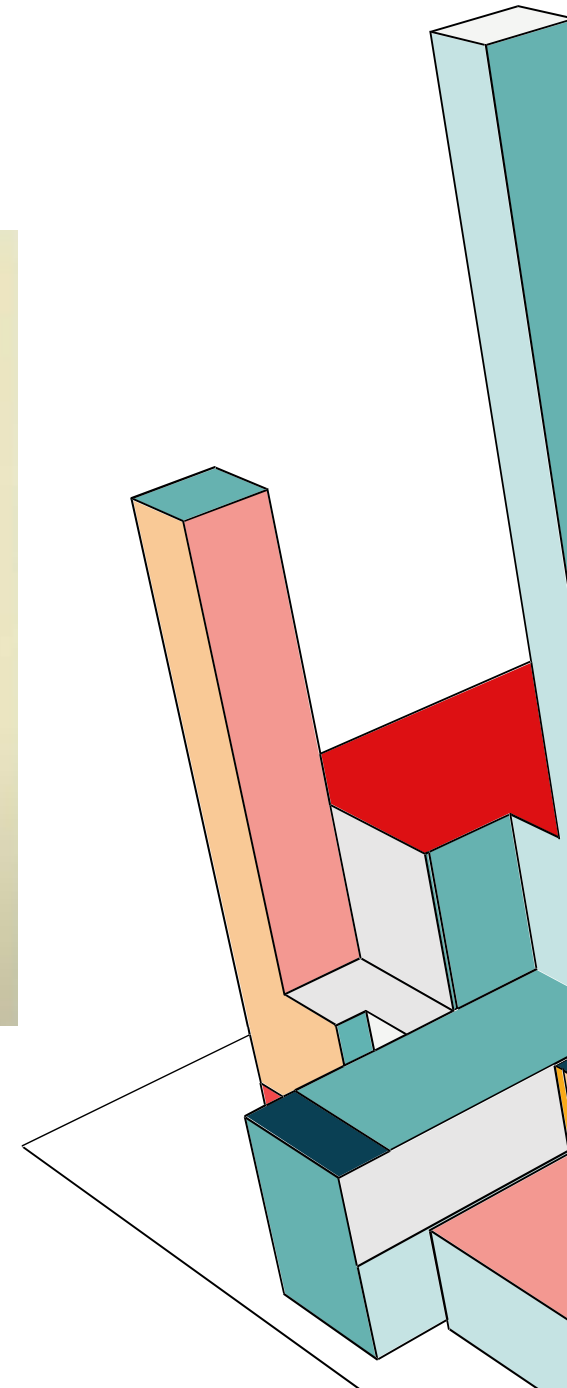
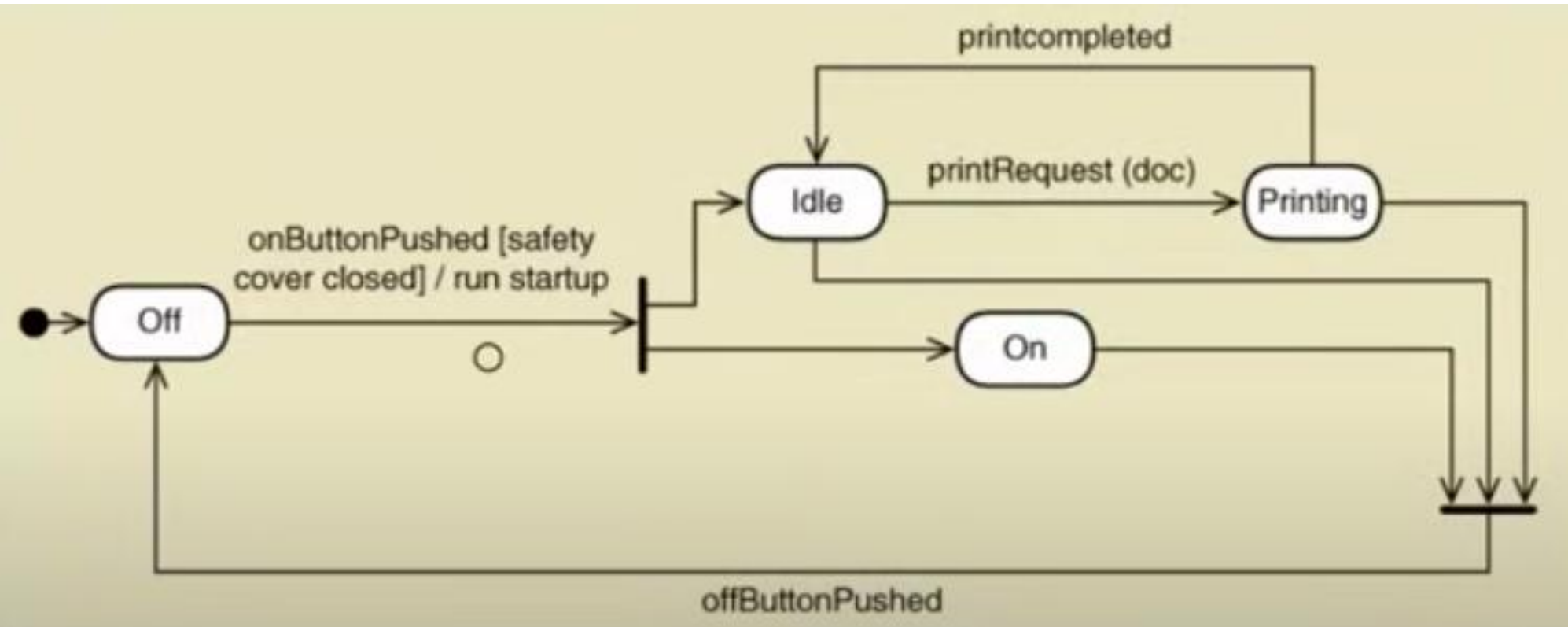
UML State Machine Diagrams

Example:



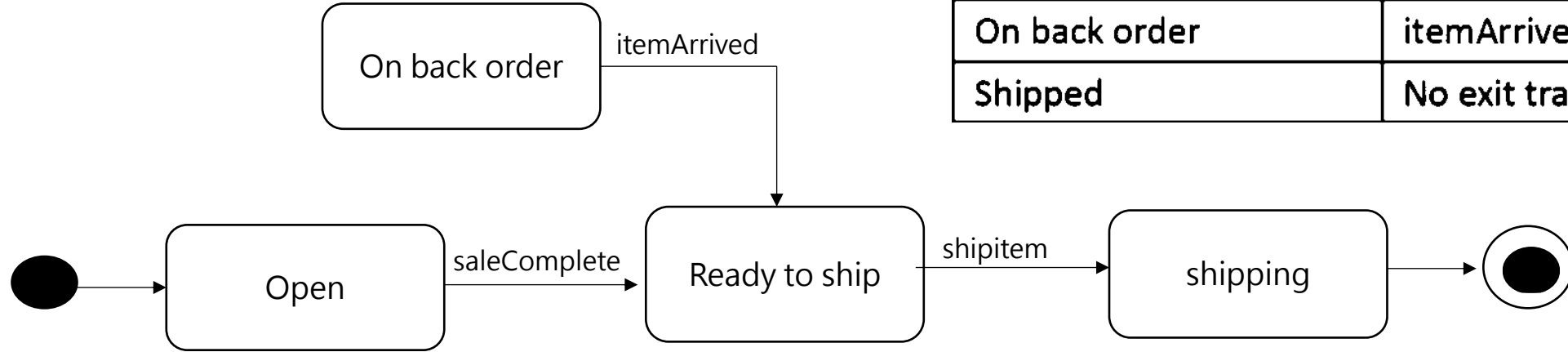
UML State Machine Diagrams

Example: Printer with Concurrent paths



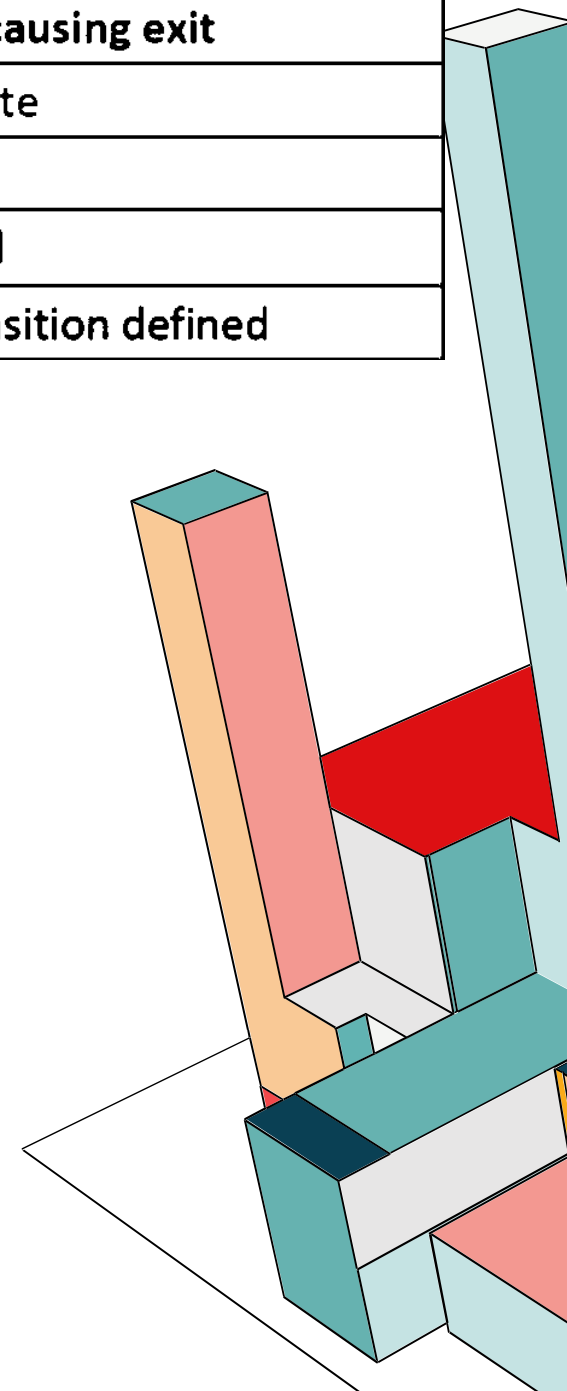
UML State Machine Diagrams

Example: SaleItem



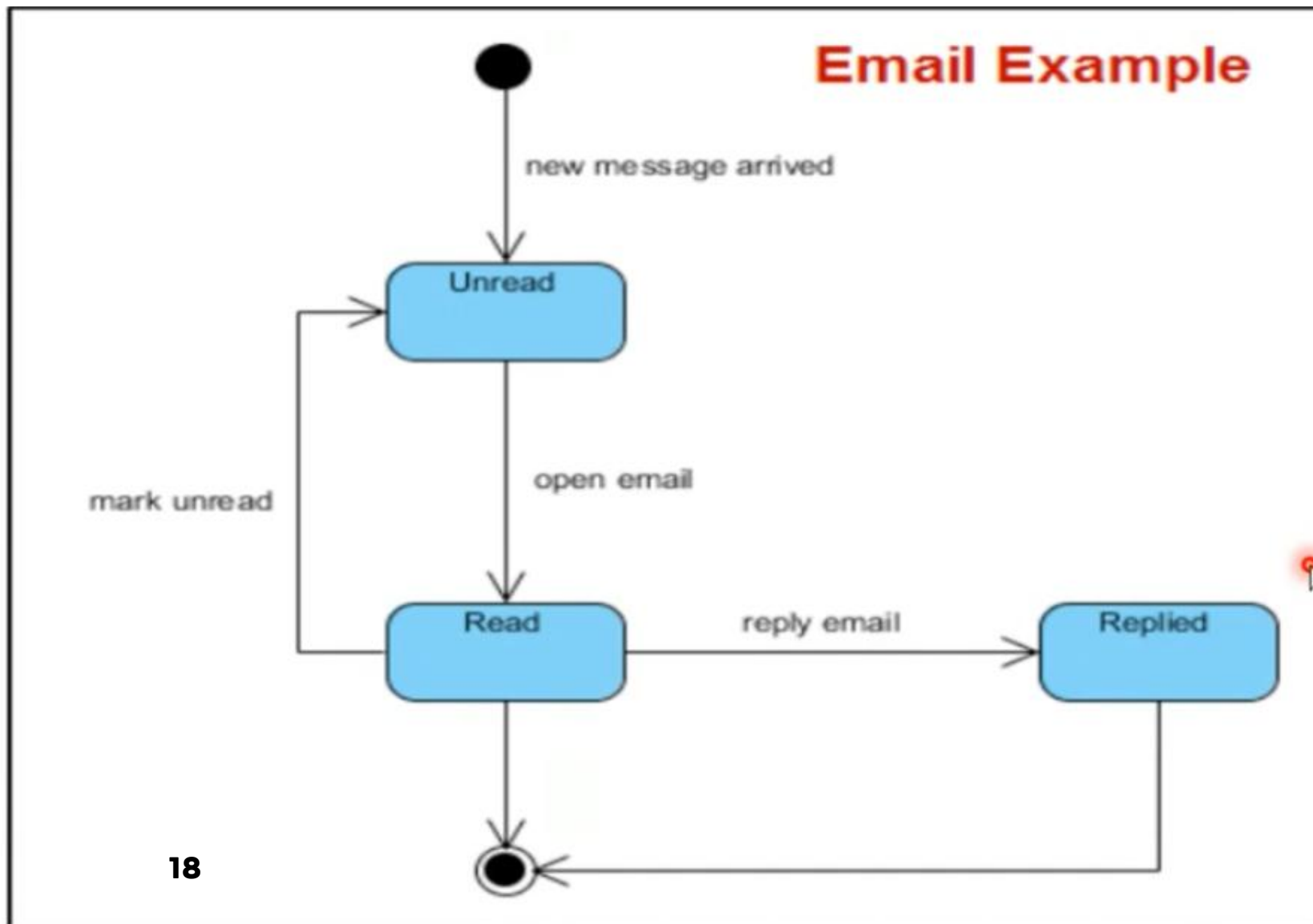
State	Transition causing exit
Open	saleComplete
Ready to Ship	shipItem
On back order	itemArrived
Shipped	No exit transition defined

هذا المخطط يوضح كيف يمكن لنفس الكائن (المنتج) أن يمر بمسارات مختلفة حسب الأحداث:
إما أن يُباع مباشرة ويصبح جاهزاً للشحن.
أو يدخل في حالة طلب مؤجل حتى وصول المخزون.
وفي كلتا الحالتين، ينتهي المسار عند الشحن وإغلاق دورة حياة المنتج.



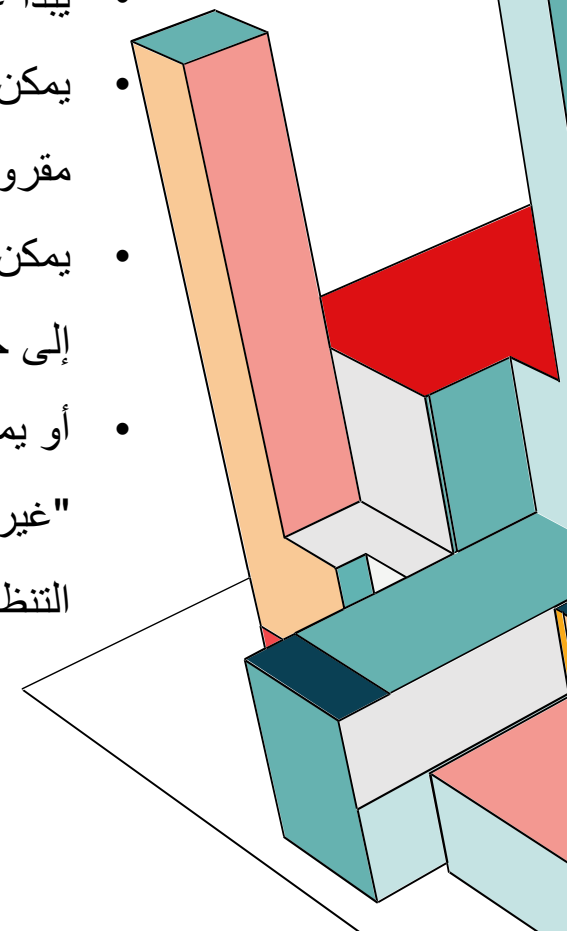
UML State Machine Diagrams

Example:



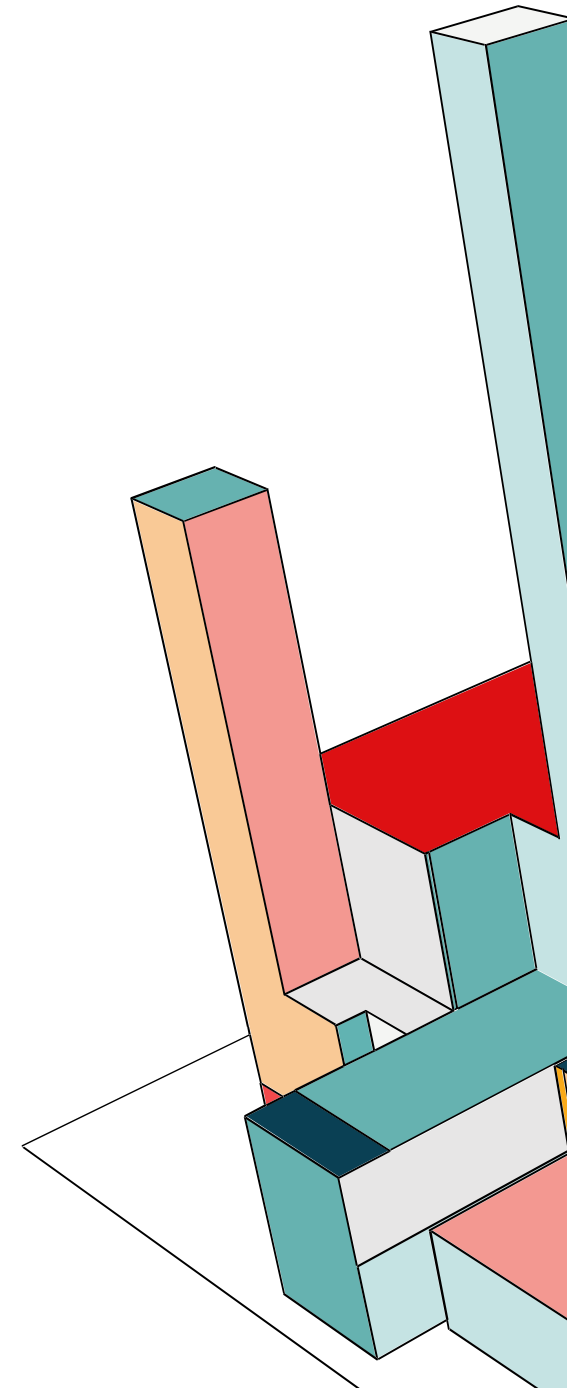
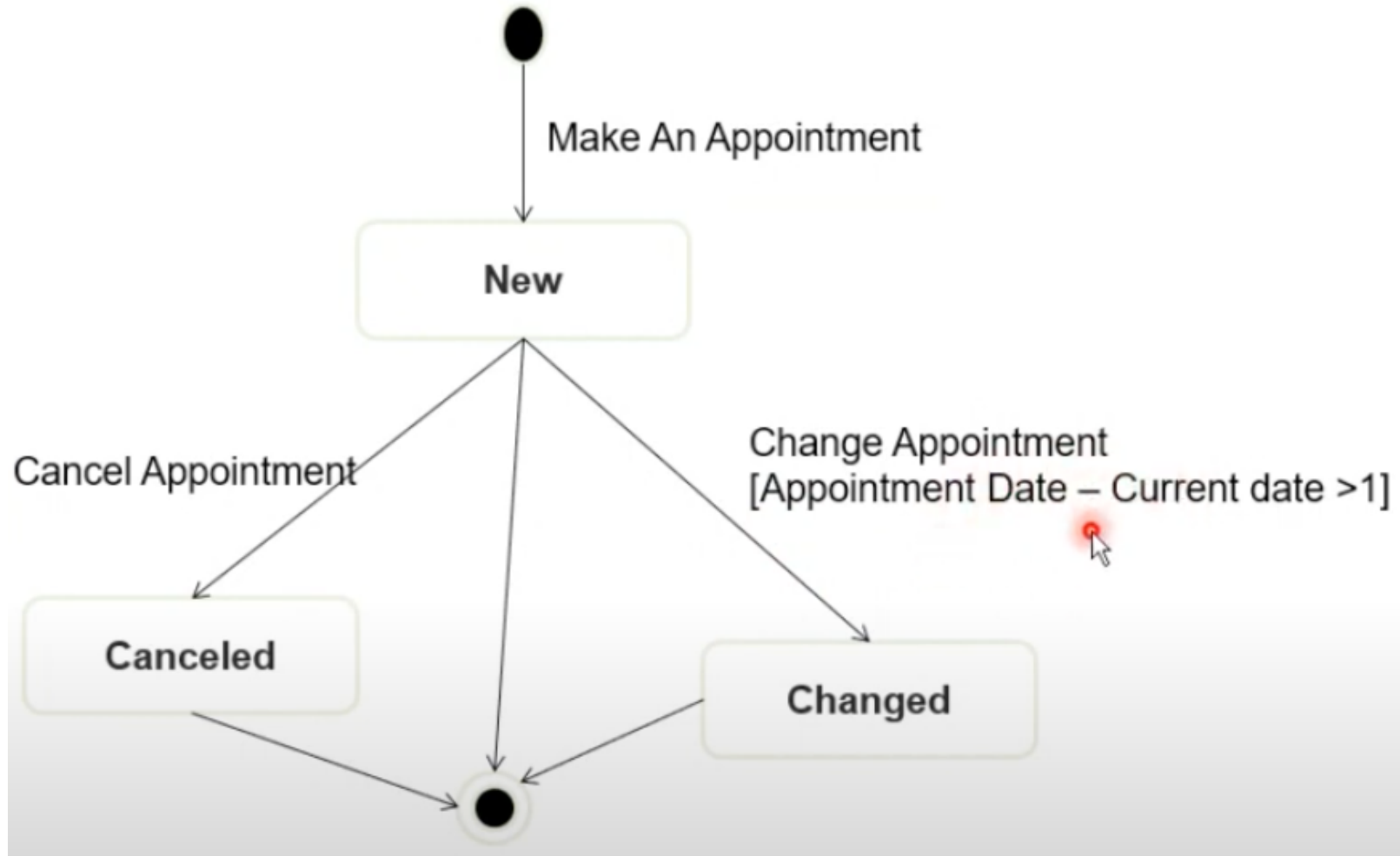
هذا المثال يوضح كيف يمكن لمخطط الحالة أن يعكس السلوك الديناميكي للبريد الإلكتروني:

- يبدأ غير مقروء.
- يمكن أن يُفتح ويصبح مقروءًا.
- يمكن أن يُرد عليه فينتقل إلى حالة "تم الرد".
- أو يمكن أن يُعاد إلى حالة "غير مقروء" لأغراض التنظيم أو التذكير.



UML State Machine Diagrams

Example:



UML Interaction diagrams

Interaction diagrams are dynamic. They describe how objects collaborate.

- They help you to visualize how the system runs.
- An interaction diagram is often built from a use case and a class diagram.
 - The objective is to show how a set of objects accomplish the required interactions with an actor.

تُعد مخططات التفاعل Interaction Diagrams

مخططات ديناميكية تُستخدم لتمثيل كيفية تعاون الكائنات

Objects فيما بينها. تساعد هذه المخططات على تصوّر

كيفية عمل النظام أثناء التشغيل بشكل عملي.

عادةً ما يتم بناء مخطط التفاعل بالاعتماد على حالة استخدام

Use Case و مخطط الفئات Class Diagram ، حيث

يهدف إلى إظهار الكيفية التي تنجز بها مجموعة من الكائنات

التفاعلات المطلوبة مع الممثل Actor .



UML Interaction diagrams

- Interaction diagrams show how a set of actors and objects communicate with each other to perform the steps of a use case.
- The set of steps, taken together, is called an *interaction*.
- Interaction diagrams can show several different types of communication.
 - E.g. method calls, messages send over the network
 - These are all referred to as *messages*.

تُظهر مخططات التفاعل Interaction Diagrams كيفية تواصل مجموعة من الممثلين Actors والكائنات Objects مع بعضهم البعض من أجل تنفيذ خطوات حالة استخدام معينة. ويُطلق على مجموعة هذه الخطوات مجتمعة اسم التفاعل Interaction. يمكن لمخططات التفاعل أن تُبرز عدة أنواع مختلفة من وسائل الاتصال، مثل:

- استدعاء الدوال Method Calls
- إرسال الرسائل عبر الشبكة Messages sent over the network

وجميع هذه الوسائل يُشار إليها بشكل عام باسم الرسائل Messages



Elements found in interaction diagrams

العناصر الموجودة في مخططات التفاعل

Instances of classes

1. مثيلات الفئات Instances of Classes

- Shown as boxes with the class and object identifier underlined

- تُعرض على شكل مربعات تحتوي على اسم الفئة ومعرّف الكائن

Object Identifier

- يتم تسطير الاسم للدلالة على أنه مثيل Instance محدد من الفئة.

Actors

2. الممثلون Actors

- Use the stick-person symbol as in use case diagrams

- يُستخدم لهم رمز الشخص العصا Stick-Person كما هو الحال في

مخططات حالات الاستخدام Use Case Diagrams .

- يمثلون الأطراف الخارجية التي تتفاعل مع النظام.

Messages





3. الرسائل Messages

- Shown as arrows from actor to object, or from object to object

- تُعرض على شكل أسهم من الممثل إلى الكائن، أو من كائن إلى آخر.

- تمثل عمليات الاتصال مثل استدعاء الدوال Method Calls أو

إرسال الرسائل عبر الشبكة.

			
Synchronous Or Call	Asynchronous	Creation	Reply (Return)



Creating interaction diagrams

You should develop a class diagram and a use case model before starting to create an interaction diagram.

There are two kinds of interaction diagrams:

- ❖ *Sequence diagrams*
- ❖ *Communication diagrams*

قبل البدء في إنشاء مخطط التفاعل ، Interaction Diagram

ينبغي أولاً تطوير كل من:

❑ مخطط الفئات Class Diagram

❑ نموذج حالة الاستخدام Use Case Model

وذلك لأن مخطط التفاعل يعتمد على هذه النماذج لتوضيح كيفية التواصل بين الكائنات والممثلين.

يوجد نوعان رئيسيان من مخططات التفاعل:

❖ مخططات التسلسل Sequence Diagram تُظهر ترتيب

الرسائل والعمليات عبر الزمن.

❖ مخططات الاتصال Communication Diagram تُبرز

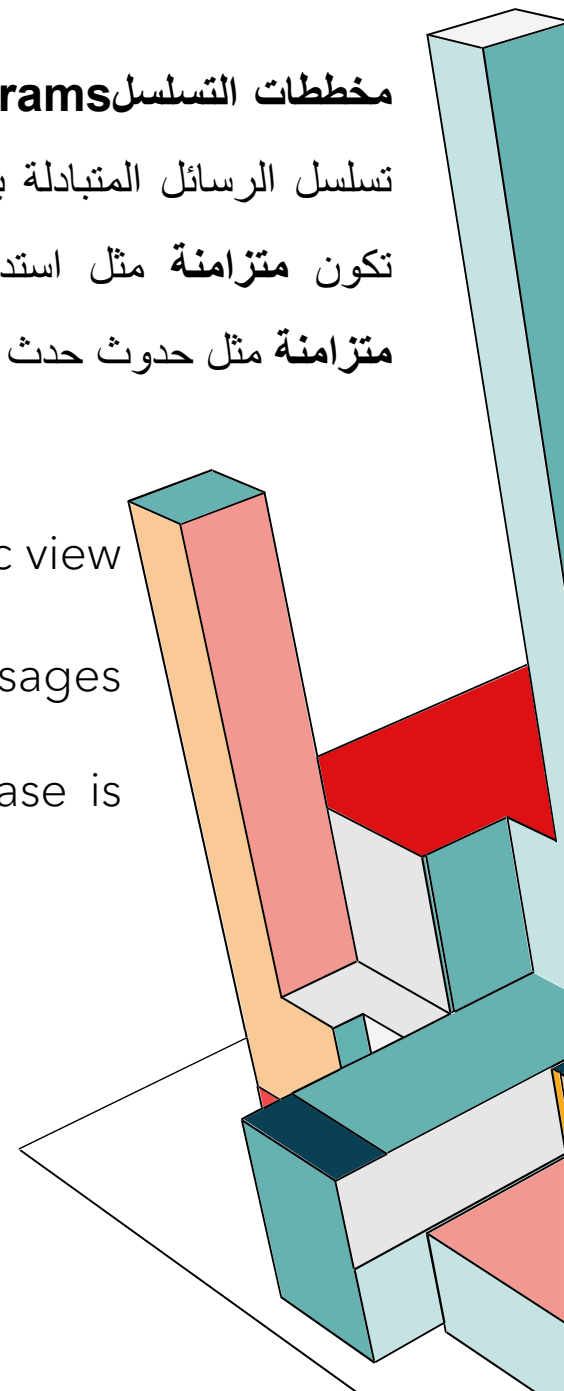
العلاقات بين الكائنات وكيفية تبادل الرسائل فيما بينها.



- **Sequence diagrams** illustrates the sequence of messages between objects in an interaction. these messages may be synchronous, as in a method call or asynchronous, as in an event.

مخططات التسلسل Sequence Diagrams تُستخدم لتوضيح تسلسل الرسائل المتبادلة بين الكائنات أثناء التفاعل. هذه الرسائل قد تكون متزامنة مثل استدعاء دالة (Method Call) ، أو غير متزامنة مثل حدوث حدث (Event)
- **A communication diagram** is a UML interaction diagram that depicts a dynamic view of a group of objects interacting with each other by showing the sequence of messages passed among them. the sequence in which the objects participate in each use case is depicted by means of message sequence numbers

مخططات الاتصال (Communication Diagrams): هي نوع من مخططات التفاعل في UML ، تُظهر رؤية ديناميكية لمجموعة من الكائنات وهي تتفاعل مع بعضها البعض. يتم توضيح تسلسل الرسائل المتبادلة بينها باستخدام أرقام تسلسل الرسائل (Message Sequence Numbers) ، مما يبين ترتيب مشاركة الكائنات في كل حالة استخدام.



مقارنة

Communication Diagram	Sequence Diagram
يوضح العلاقات بين الكائنات وكيفية تبادل الرسائل	يوضح ترتيب الرسائل والعمليات عبر الزمن
التركيز على البنية والعلاقات بين الكائنات	التركيز على التسلسل الزمني للرسائل
مربعات للكائنات وروابط بينها، والرسائل تظهر على الروابط	خط زمني عمودي لكل كائن، والرسائل تظهر كسهام أفقية
مناسب لتوضيح شبكة التفاعلات بين الكائنات	مناسب لتوضيح الخطوات المتتابعة في حالة استخدام
يعطي صورة أوضح عن العلاقات بين الكائنات	يعطي صورة دقيقة عن ترتيب الأحداث

العنصر

الهدف

التركيز

التمثيل البصري

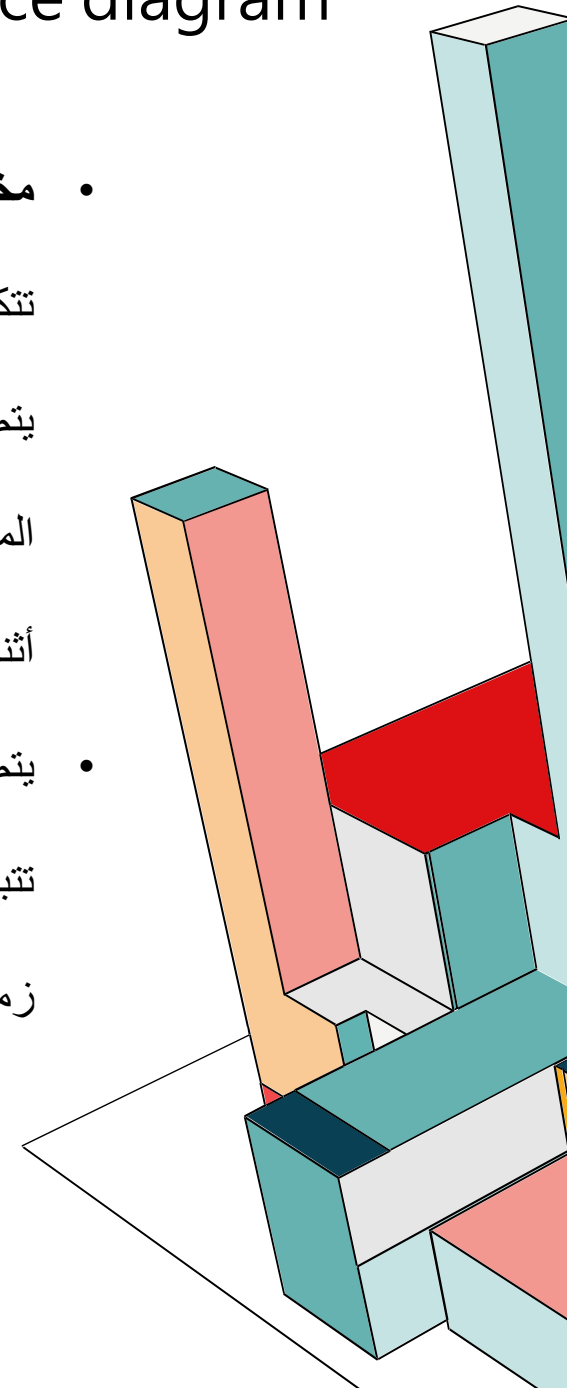
الاستخدام الشائع

المزايا

- إذا أردت أن تركز على متى تحدث الرسائل والخطوات → استخدم Sequence Diagram.
- إذا أردت أن تركز على من يتواصل مع من وكيفية الترابط بينهم → استخدم Communication Diagram.
- إذا أردت التركيز على الترتيب الزمني للرسائل → استخدم Sequence Diagram.
- إذا أردت التركيز على العلاقات والارتباطات بين الكائنات → استخدم Communication Diagram.
- كلا المخططين يُظهران تسلسل الرسائل، لكن كل واحد منهما يبرز جانبًا مختلفًا من التفاعل.

- **Sequence diagrams**
- consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.
- Sequence diagrams are organized according to time.

- Sequence Diagrams **مخططات التسلسل**
تتكون من مجموعة من الكائنات Objects التي يتم تمثيلها بخطوط حياة Lifelines. تُظهر هذه المخططات الرسائل المتبادلة بين الكائنات عبر الزمن أثناء التفاعل.
- يتم تنظيم مخططات التسلسل وفقًا للزمن، بحيث يمكن تتبع ترتيب الرسائل والعمليات خطوة بخطوة بشكل زمني واضح.

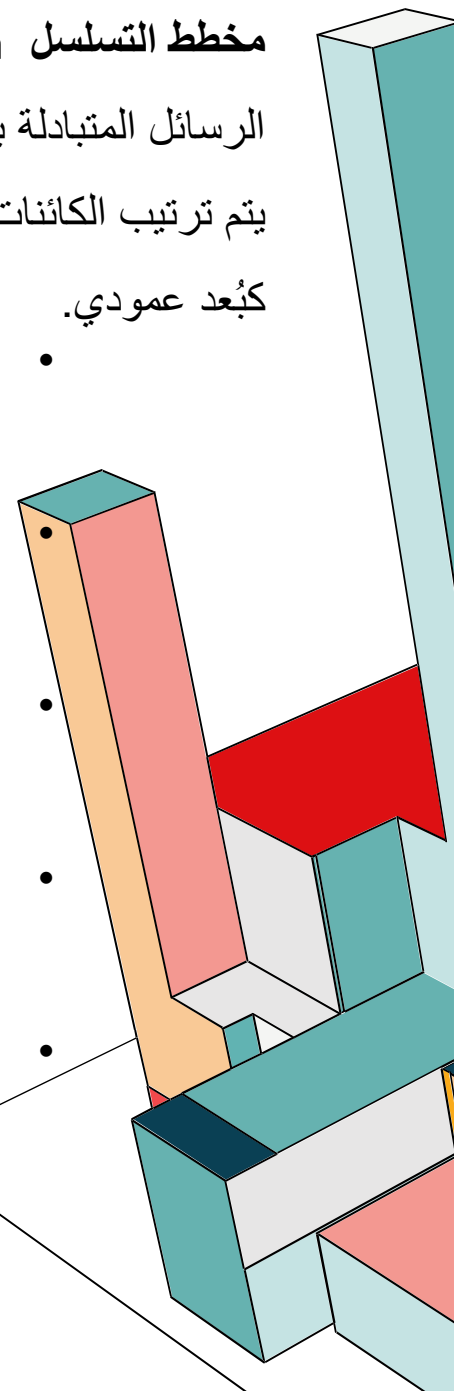


A **sequence diagram** shows the sequence of messages exchanged by the set of objects performing a certain task

- The objects are arranged horizontally across the diagram.
- An actor that initiates the interaction is often shown on the left.
- The vertical dimension represents time.
- A vertical line, called a *lifeline*, is attached to each object or actor.
- The lifeline becomes a broad box, called an *activation box* during the *live activation* period.
- A message is represented as an arrow between activation boxes of the sender and receiver.
 - A message is labelled and can have an argument list and a return value.

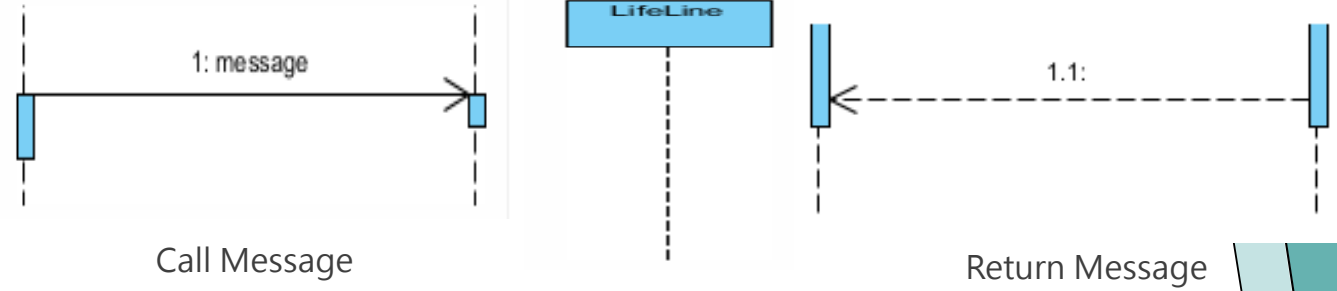
مخطط التسلسل Sequence Diagram يوضح تسلسل الرسائل المتبادلة بين مجموعة من الكائنات أثناء تنفيذ مهمة معينة. يتم ترتيب الكائنات بشكل أفقي عبر المخطط، بينما يُعرض الزمن كبُعد عمودي.

- غالبًا ما يُوضع الممثل Actor الذي يبدأ التفاعل على الجانب الأيسر من المخطط.
- يُرسم لكل كائن أو ممثل خط عمودي يسمى **خط الحياة** Lifeline.
- أثناء فترة التفعيل، يتحول خط الحياة إلى مربع عريض يسمى **صندوق التفعيل** Activation Box.
- تُعرض الرسائل على شكل أسهم بين صناديق التفعيل الخاصة بالمرسل والمستقبل.
- كل رسالة تكون مُسمّاة، ويمكن أن تحتوي على قائمة معاملات Argument List وقيمة إرجاع Return Value.



The key components and concepts of sequence diagrams:

- **Objects:** Objects represent the instances of classes or components participating in the interaction.
- **Lifelines:** Lifelines visually depict the existence of an object over a period of time.
- **Messages:** Messages represent the communication or interaction between objects. Messages can be:
 - **Return Message** Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.



الكائنات تمثل مثيلات الفئات أو المكونات المشاركة في التفاعل. تُعرض عادةً في المخطط كعناصر أفقية في الأعلى.

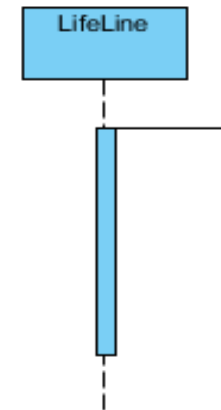
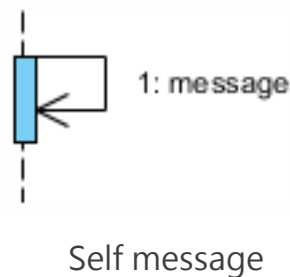
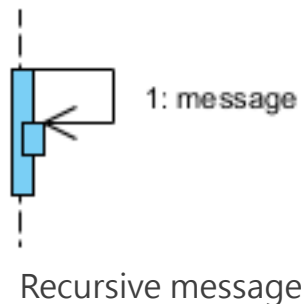
خطوط الحياة Lifelines تُظهر بشكل بصري وجود الكائن خلال فترة زمنية معينة. تُرسم كخطوط عمودية تمتد من الكائن إلى الأسفل، مما يوضح استمرار وجوده أثناء التفاعل.

الرسائل Messages تمثل عملية الاتصال أو التفاعل بين الكائنات. تُعرض على شكل أسهم بين خطوط الحياة الخاصة بالمرسل والمستقبل. يمكن أن تكون الرسائل:

- رسائل عادية مثل استدعاء دالة أو إرسال حدث.
- رسائل إرجاع Return Messages وهي نوع خاص من الرسائل يُستخدم لتمثيل تمرير المعلومات مرة أخرى إلى المستدعي استجابةً لرسالة سابقة.

The key components and concepts of sequence diagrams:

- **Recursive message** is a kind of message that represents the invocation of message of the same lifeline
- **Activation Bars**: Activation bars show the period of time during which an object is actively processing a message.



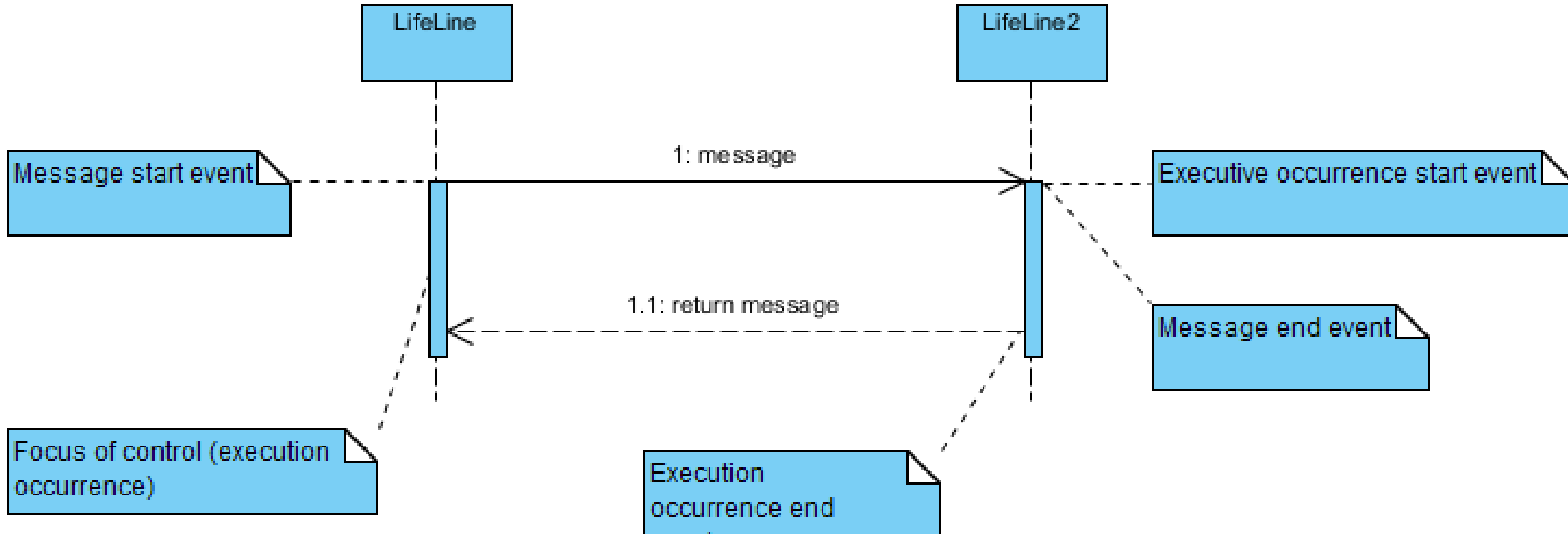
UML Sequence diagram

الرسالة التكرارية Recursive Message هي نوع من الرسائل يُستخدم لتمثيل استدعاء رسالة من نفس خط الحياة Lifeline الخاص بالكائن. بمعنى أن الكائن يقوم باستدعاء عملية أو وظيفة خاصة به بشكل متكرر أثناء التفاعل.

أشرطة التفعيل Activation Bars تُظهر الفترة الزمنية التي يكون فيها الكائن نشطًا ويقوم بمعالجة رسالة معينة. تُرسم كصندوق مستطيل ضيق على خط الحياة، يبدأ عند استقبال الرسالة وينتهي عند اكتمال معالجتها. تساعد في توضيح مدة التنفيذ لكل عملية يقوم بها الكائن.

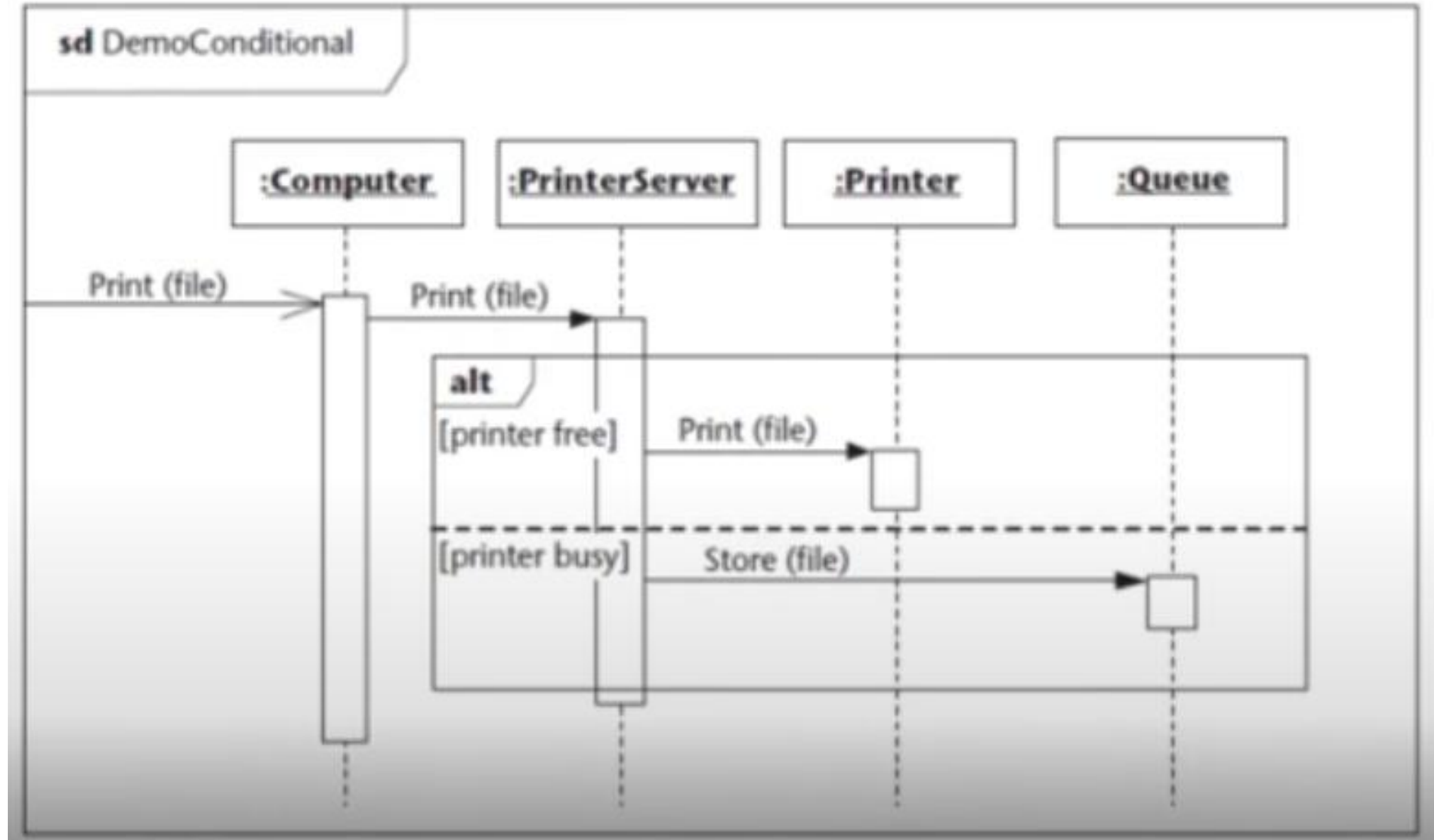


UML Sequence diagram



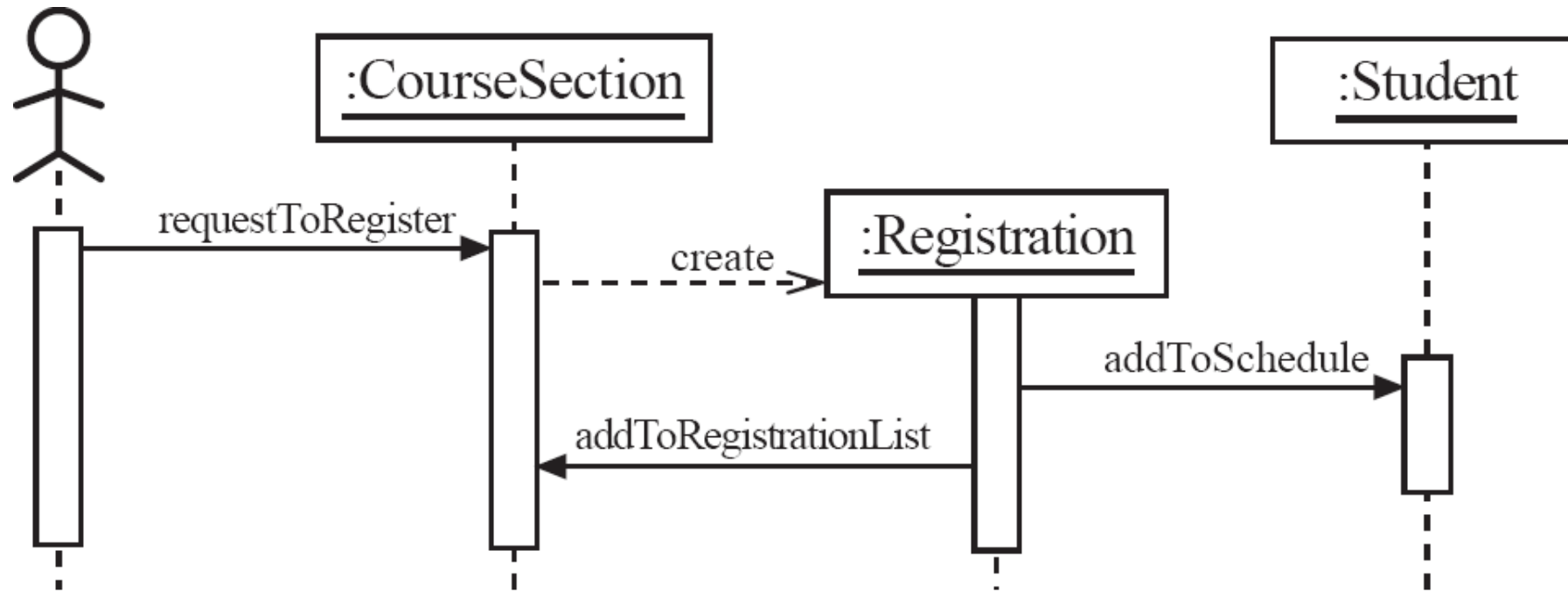
UML Sequence diagram

Sequence diagrams – an example

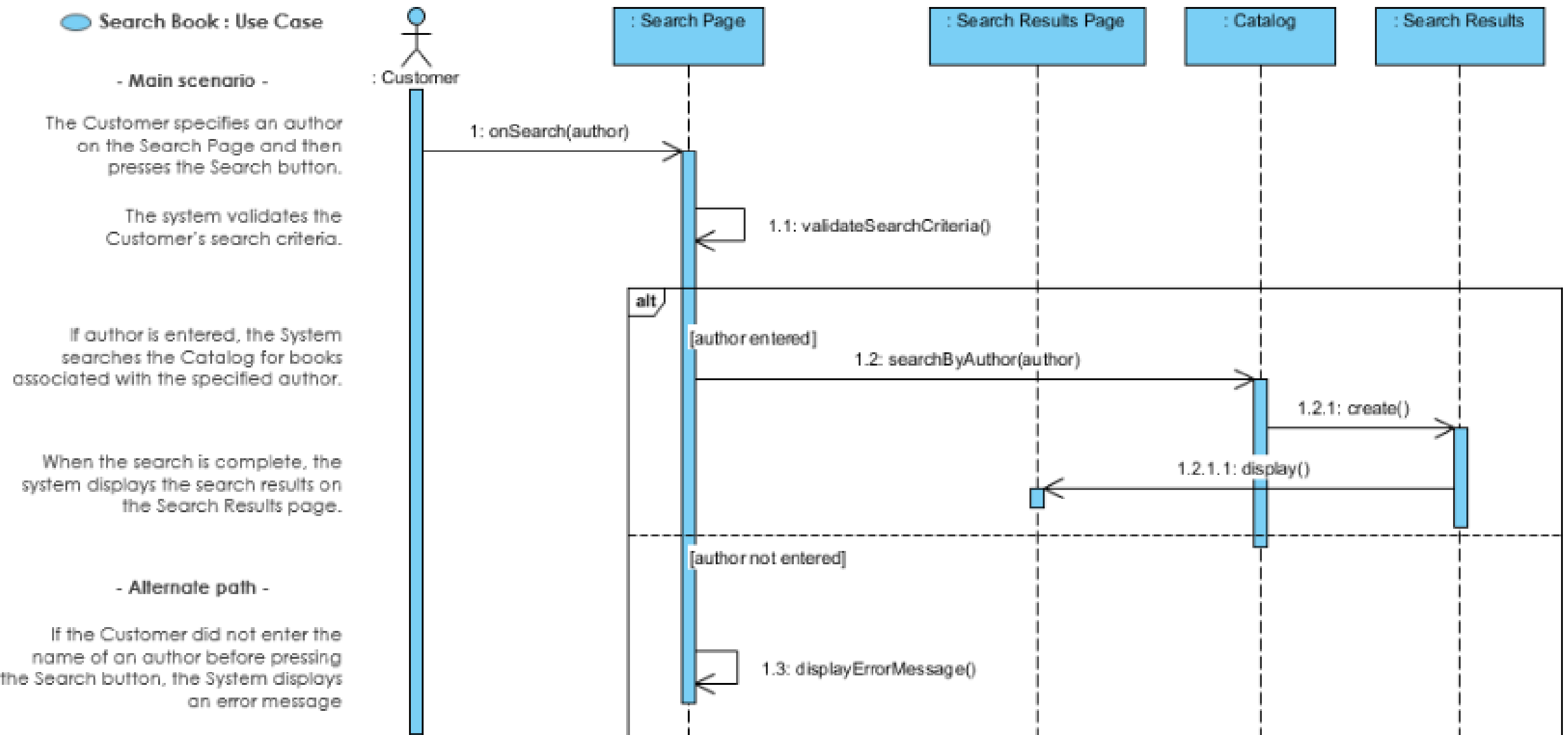


Sequence diagrams – an example

UML Sequence diagram



Sequence Diagram Example 1: Library System, Search book



UML communication diagram

• مخطط الاتصال Communication Diagram هو أحد

مخططات التفاعل في UML، ويُستخدم لتصوير رؤية ديناميكية

لمجموعة من الكائنات وهي تتفاعل مع بعضها البعض من خلال

إظهار تسلسل الرسائل المتبادلة بينها.

• في مخطط الاتصال، يتم توضيح ترتيب مشاركة الكائنات في كل حالة

استخدام باستخدام أرقام تسلسل الرسائل Message Sequence

Numbers. ويجب أن يتوافق هذا التسلسل في المخطط مع تسلسل

التفاعلات بين الممثل Actor والنظام كما هو موصوف مسبقًا في

حالة الاستخدام Use Case.



- A communication diagram is a UML interaction diagram that depicts a dynamic view of a group of objects interacting with each other by showing the sequence of messages passed among them.
- On a communication diagram, the sequence in which the objects participate in each use case is depicted by means of message sequence numbers. The message sequencing on the communication diagram should correspond to the sequence of interactions between the actor and the system already described in the use case.

- Communication diagrams focus both on the interactions and the links among a set of collaborating objects.
- Communication diagrams show objects and how messages are sent between the linked objects and indicate their relations.

❖ The key components and concepts of communication diagrams:

1. An actor:

- Is a person or system that derives benefit from and is external to the system.
- Participates in a collaboration by sending and/or receiving messages.
- Is depicted either as a stick figure (default) or, if a nonhuman actor is involved, as a rectangle with <<actor>> in it (alternative)

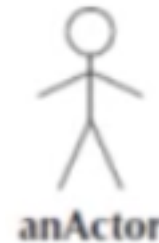
UML communication diagram

تركز مخططات الاتصال على التفاعلات وكذلك على الروابط بين مجموعة من الكائنات المتعاونة. فهي تُظهر الكائنات، وكيف يتم تبادل الرسائل بينها عبر الروابط، مع توضيح العلاقات التي تربطها.

1. الممثل Actor هو شخص أو نظام خارجي يستفيد من النظام.

- يشارك في التعاون من خلال إرسال واستقبال الرسائل.
- يُعرض عادةً برمز الشخص العصا Stick Figure ، أو إذا كان الممثل غير بشري يُعرض كمستطيل يحتوي

على الوسم <<actor>>



An object:

- Participates in a collaboration by sending and/or receiving messages.

anObject : aClass

An association:

- Shows an association between actors and/or objects.
- Is used to send messages.

Seq Number: aMessage →

A message:

- Conveys information from one object to another one.
- Has direction shown using an arrowhead.
- Has sequence shown by a sequence number.

UML communication diagram

2. الكائن Object يمثل مثيلاً من فئة معينة.

- يشارك في التعاون من خلال إرسال واستقبال الرسائل مع الممثلين أو الكائنات الأخرى.

3. الارتباط Association يُظهر العلاقة بين الممثلين والكائنات.

- يُستخدم كقناة لإرسال الرسائل بينهم.

4. الرسالة Message تنقل المعلومات من كائن إلى آخر.

- تُعرض كسهم يوضح اتجاه الرسالة من المرسل إلى المستقبل.

يُحدد ترتيب الرسائل باستخدام أرقام تسلسل Sequence

Numbers لتوضيح التسلسل الزمني للتفاعل.



A message:

A message label (the label placed on a message) in a communication diagram is specified with the syntax:

- predecessor guard-condition sequence-expression

return-value := signature

The predecessor is specified with the syntax:

sequence-number ',' ... '/'

The predecessor is an expression for synchronization of threads or paths, meaning that the messages connected to specified sequence-numbers must be performed and handled before the current message is sent (that is, execution is continued). The list of sequence-numbers is comma separated.

Eg: 1: display ()

Eg: 1.1a,1.1b/1.2: continue()

UML communication diagram

يتم تحديد تسمية الرسالة في مخطط الاتصال باستخدام البنية التالية:

predecessor guard-condition sequence-expression

return-value := signature

Predecessor المُسبق: يُكتب بالصياغة التالية:

sequence-number ',' ... '/'

وهو تعبير يُستخدم لمزامنة الخيوط (Threads) أو المسارات (Paths) ، بحيث يجب تنفيذ ومعالجة الرسائل المرتبطة بأرقام التسلسل المحددة قبل إرسال الرسالة الحالية (أي قبل متابعة التنفيذ). تُكتب أرقام التسلسل مفصولة بفواصل.

◆ مثال 1:

1: display()

هنا الرسالة الأولى تحمل الرقم (1) وتقوم بتنفيذ العملية display().

• مثال 2:

1.1a,1.1b/1.2: continue()

في هذا المثال:

- يجب تنفيذ الرسائل 1.1a و 1.1b قبل الرسالة 1.2.
- بعد ذلك يتم إرسال الرسالة continue(). هذا يوضح كيفية التحكم في ترتيب الرسائل ومزامنتها بين الكائنات.



UML communication diagram

Message Labels: Guard-condition

The guard-condition is specified with the syntax:

'[condition-clause]'

The condition-clause is normally expressed in pseudocode or an actual programming language.

Eg: [mode = display] 1.2.3.7: redraw()

شرط الحراسة (**Guard-condition**) هو شرط منطقي يُستخدم لتحديد ما إذا كان يمكن إرسال الرسالة أم لا. تتم كتابته داخل أقواس مربعة وفق الصياغة التالية:

[condition-clause]

حيث أن condition-clause هو عبارة عن جملة شرطية تُكتب عادةً بلغة شبه برمجية (Pseudocode) أو بلغة برمجة فعلية.

♦ مثال توضيحي

[mode = display] 1.2.3.7: redraw()

هنا الشرط هو mode = display :

أي أن الرسالة redraw() ذات الرقم التسلسلي 1.2.3.7 لن تُرسل إلا إذا كان النظام في وضع العرض. (display mode)

استخدام Guard-condition في مخططات الاتصال

يتيح التحكم في تدفق الرسائل بحيث يتم إرسالها فقط

عند تحقق شرط معين، مما يجعل المخطط أكثر دقة في

تمثيل منطق النظام.



UML communication diagram

الصياغة العامة يُكتب التعبير التسلسلي وفق البنية التالية:

[integer | name] [recurrence] ':'

- The sequence-expression has the syntax:
[integer | name] [recurrence] ':'
- The integer is a sequence-number specifying the message order.
- Message 1 always starts a sequence of messages; message 1.1 is the first nested message within the handling of message 1; message 1.2 is the second nested message within the handling of message 1.
- An example sequence is message 1, message 1.1, 1.2, 1.2.1, 1.2.2, 1.3, and so on.
- Thus, the numbering can describe both the sequence and nesting of messages (when the messages are synchronous, nested operation invocations and their replies).
- The name represents a concurrent thread of control; for instance, 1.2a and 1.2b are concurrent (asynchronous) messages sent in parallel.
- The sequence expression should be terminated with a colon (:).

- **العدد الصحيح (Integer):** يمثل رقم التسلسل الذي يحدد ترتيب الرسائل.
- **الاسم (Name):** يُستخدم لتمثيل خيط تحكم متزامن أو متوازي (Concurrent Thread of Control).
- **التكرار (Recurrence):** يوضح إذا كانت الرسالة متكررة أو متزامنة.
- يجب أن يُختتم التعبير بعلامة النقطتين (:).

- الرسالة 1 تبدأ دائماً تسلسل الرسائل.
- الرسالة 1.1 هي أول رسالة متداخلة ضمن معالجة الرسالة 1.
- الرسالة 1.2 هي الرسالة الثانية المتداخلة ضمن معالجة الرسالة 1.
- يمكن أن يتفرع التسلسل إلى مستويات أعمق مثل:
 - 1.2.1
 - 1.2.2
- ثم يعود إلى الرسائل التالية مثل 1.3 وهكذا.
- بهذا الشكل، يمكن للترقيم أن يصف التسلسل وكذلك التداخل بين الرسائل، خاصةً عندما تكون الرسائل متزامنة (مثل استدعاءات العمليات المتداخلة وردودها).

الرسائل المتزامنة: (Synchronous) تُظهر التداخل مثل 1 → 1.1 → 1.2 → 1.2.1 → 1.2.2 → 1.3.

الرسائل المتوازية: (Concurrent/Asynchronous) يُستخدم الاسم لتمثيل خيوط متوازية، مثل:

1.2a و 1.2b رسائل متزامنة تُرسل بالتوازي.



Message Labels: Sequence-expression UML communication diagram

- The condition-clause is normally used for specifying branches, not for guard-conditions.
- $[x < 0]$ and $[x \geq 0]$ are two condition-clauses that can be used for branching, in which **only one of the conditions is true**; thus, only one of the branches is executed (sending the message connected to that branch).
- Both condition-clauses and iteration-clauses are meant to be expressed in pseudocode or in the syntax of an actual programming language.

○ Eg. 3.1 $[x < 0]: \text{foo}()$

○ Eg. 3.2 $[x \geq 0]: \text{bar}()$

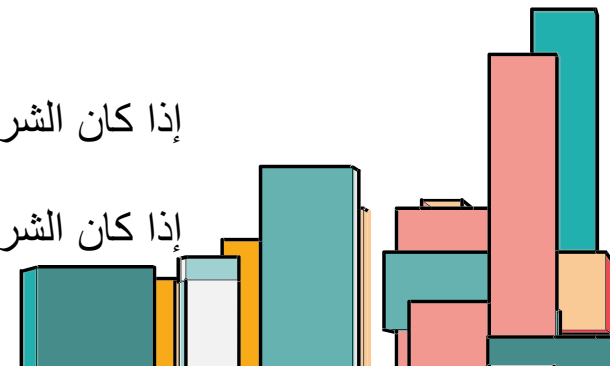
- التعبير التسلسلي مع الشرط يُستخدم لتوضيح التفرعات المنطقية في مخططات الاتصال.
- يساعد في تمثيل منطق النظام بشكل أدق، حيث يتم إرسال الرسائل وفقًا لنتائج الشروط المحددة.
- يُستخدم التعبير الشرطي (**Condition-clause**) عادةً لتحديد التفرعات (**Branches**) وليس للحراسة (Guard-conditions).
 - مثال على ذلك:
 - $[x < 0]$
 - $[x \geq 0]$
 - في هذه الحالة، يكون أحد الشرطين صحيحًا فقط، وبالتالي يتم تنفيذ فرع واحد فقط (أي إرسال الرسالة المرتبطة بذلك الفرع).
- يمكن كتابة هذه الشروط بلغة شبه برمجية (Pseudocode) أو بلغة برمجة فعلية.
- تُستخدم لتوضيح منطق التفرع داخل مخطط الاتصال.

3.1 $[x < 0]: \text{foo}()$

إذا كان الشرط $x < 0$ صحيحًا، يتم إرسال الرسالة **foo()** ذات الرقم التسلسلي 3.1

3.2 $[x \geq 0]: \text{bar}()$

إذا كان الشرط $x \geq 0$ صحيحًا، يتم إرسال الرسالة **bar()** ذات الرقم التسلسلي 3.2



UML communication diagram

- Recurrence represents a conditional or iterative execution.
يمثل التكرار (Recurrence) تنفيذًا شرطيًا أو تكراريًا للرسائل داخل مخطط الاتصال. يوجد خياران رئيسيان لكتابته:
1. [iteration-clause] *

- Two choices exist:

➤ `'*'[iteration-clause]'`

• حيث يكون **iteration-clause** هو الشرط الذي يحدد عدد مرات التكرار، مثل [i: 1...n]

➤ `'[condition-clause]'`

2. [condition-clause]

- The iteration-clause is used to specify iteration (repeated execution), where the iteration-clause is a condition for the iteration, such as [i: 1...n].

• يُستخدم لتحديد التنفيذ الشرطي. (Conditional Execution)

• حيث يتم إرسال الرسالة فقط إذا تحقق الشرط المحدد.

مثال على التكرار

1.1 `*[x = 1..10]: x : doSomething(x)`

• الرقم التسلسلي للرسالة هو 1.1

- For example, a message label containing an iteration can be shown as:

• الرمز * مع الشرط [x = 1..10] يعني أن الرسالة ستُنفذ بشكل متكرر من x = 1 إلى x = 10.

• الرسالة هي doSomething(x)، أي أنه سيتم استدعاؤها عشر مرات مع قيم مختلفة لـ x

➤ `1.1 *[x = 1..10]: x : doSomething(x)`

التكرار (Iteration) يُستخدم لتوضيح الرسائل التي تُنفذ عدة مرات وفق شرط محدد.

• التنفيذ الشرطي (Conditional Execution) يُستخدم لتوضيح الرسائل التي تُنفذ فقط عند تحقق شرط معين.

• هذه الآليات تجعل مخططات الاتصال أكثر دقة في تمثيل منطق النظام، خاصةً في الحالات التي تتطلب تكرار أو تفريع في التفاعل.



UML communication diagram

يجب أن تُسند القيمة المرجعة (Return-value) إلى توقيع الرسالة (Message Signature).

توقيع الرسالة (Signature) يتكون من:

1. اسم الرسالة (Message-name)

2. قائمة المعاملات (Argument List)

القيمة المرجعة تُظهر النتيجة التي يتم الحصول عليها من استدعاء عملية (Operation Call) أو رسالة.

مثال 1

1.4.5: x := calc(n)

- الرقم التسلسلي للرسالة هو 1.4.5.
- العملية المستدعاء هي calc(n).
- القيمة المرجعة من العملية تُسند إلى المتغير x.

مثال 2

كتابة تعليمات برمجية

2* [n: 1..z]: prim := nextPrim(prim)

- الرقم التسلسلي للرسالة هو 2* مع شرط التكرار [n: 1..z].
- العملية المستدعاء هي nextPrim(prim).
- القيمة المرجعة تُسند إلى المتغير prim في كل دورة من التكرار.

• التوقيع يوضح اسم الرسالة والمعاملات.

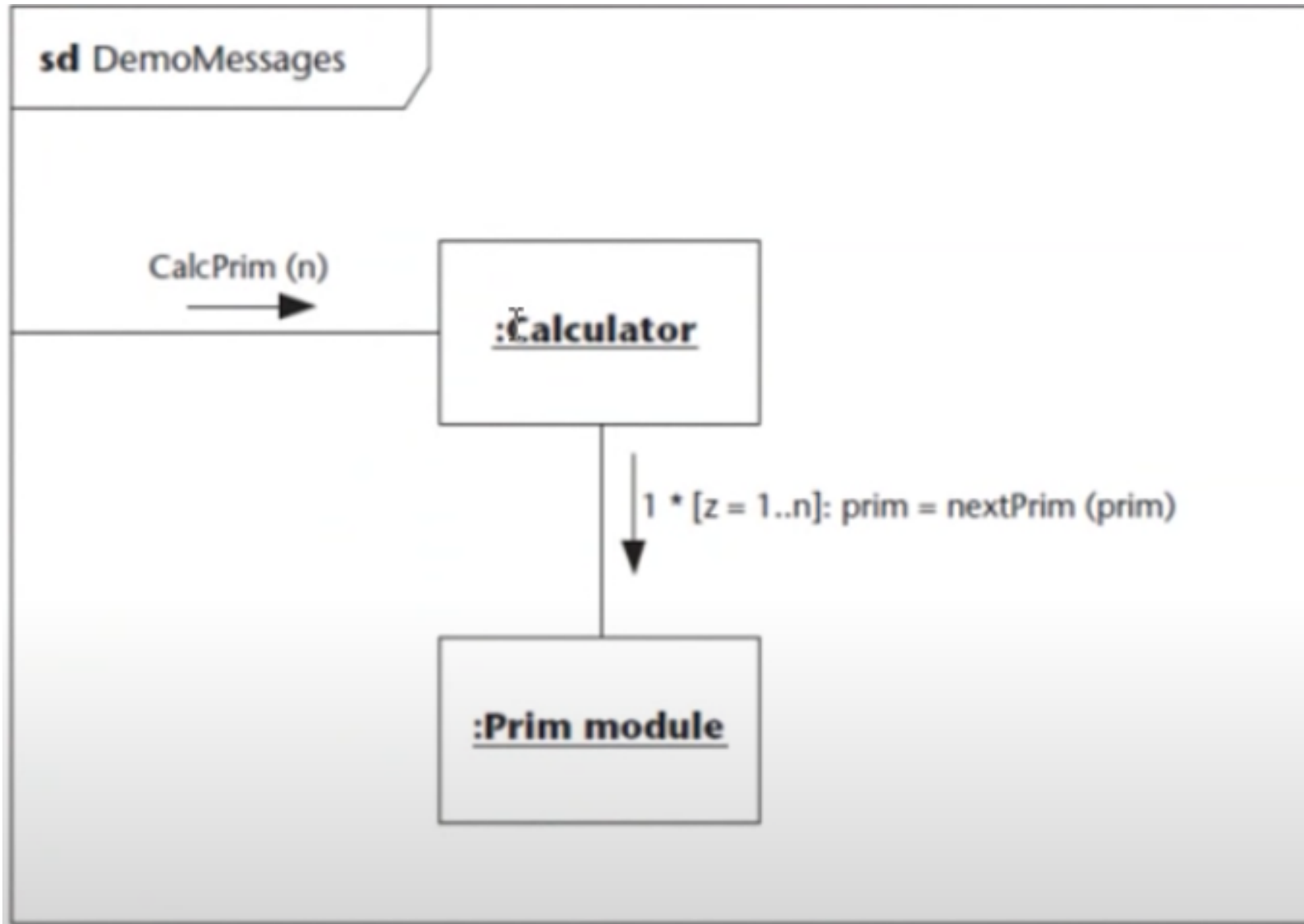
• القيمة المرجعة تُستخدم لتخزين نتيجة العملية المستدعاء.

• هذا الأسلوب يجعل مخططات الاتصال أكثر دقة في تمثيل منطق

التنفيذ وتدفق البيانات بين الكائنات.

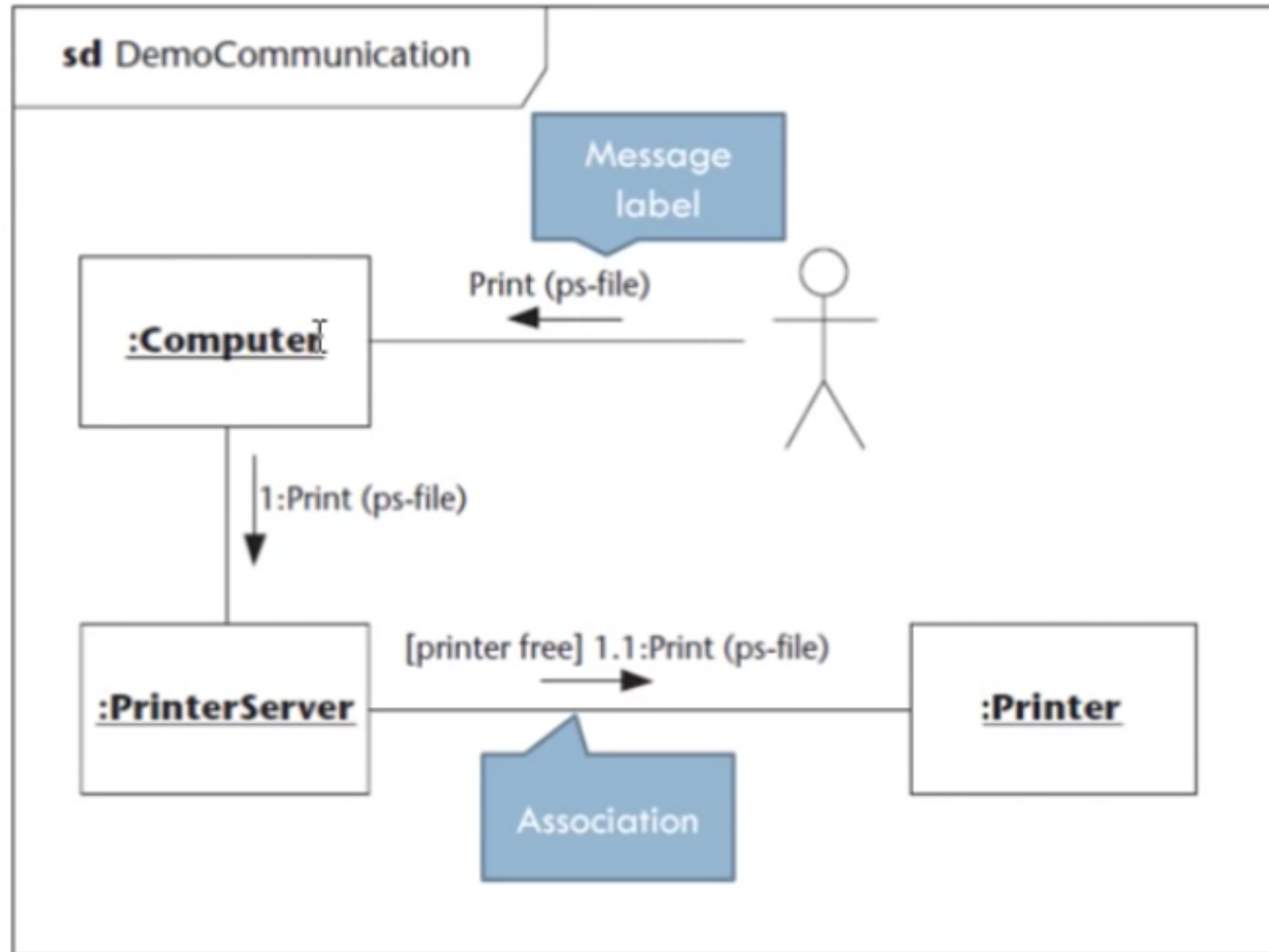


UML communication diagram



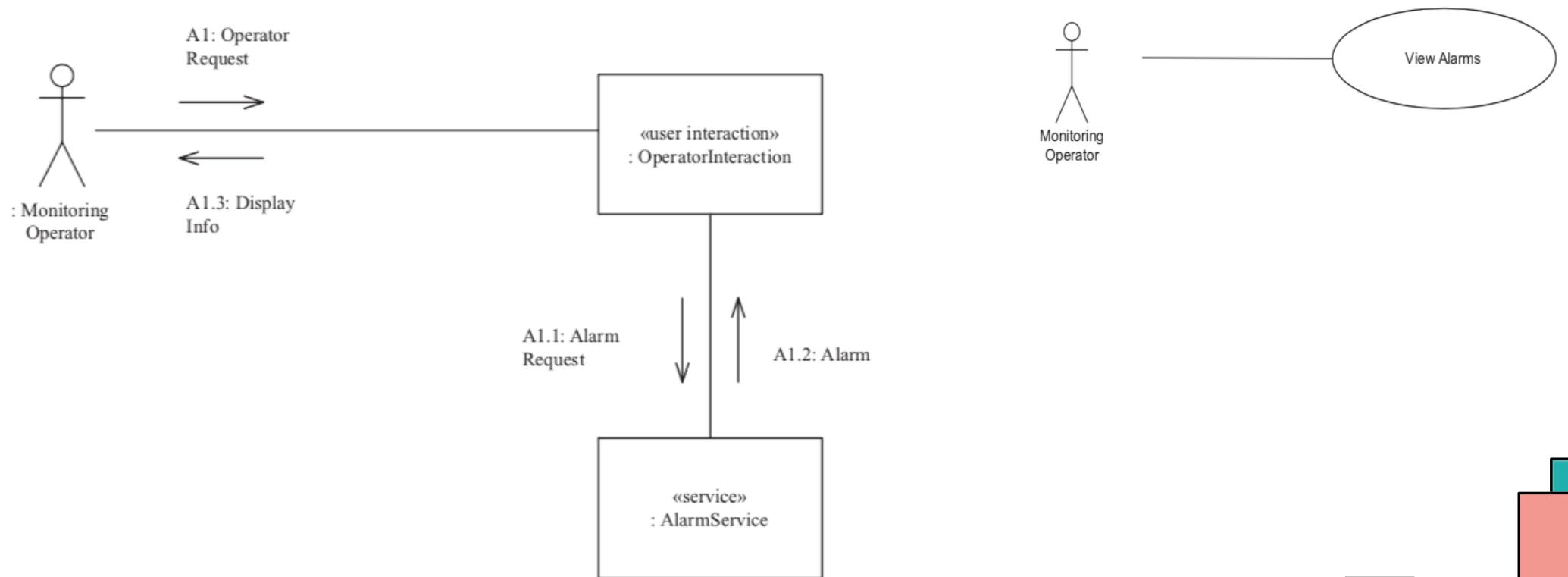
UML communication diagram

Example 1:



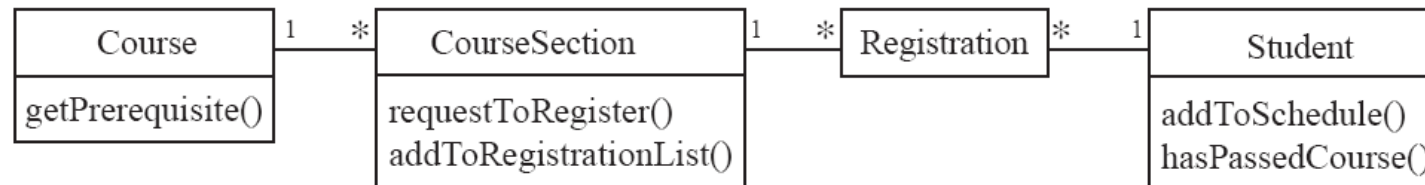
UML communication diagram

Example 2: consider the View Alarms use case from the Emergency Monitoring System case study



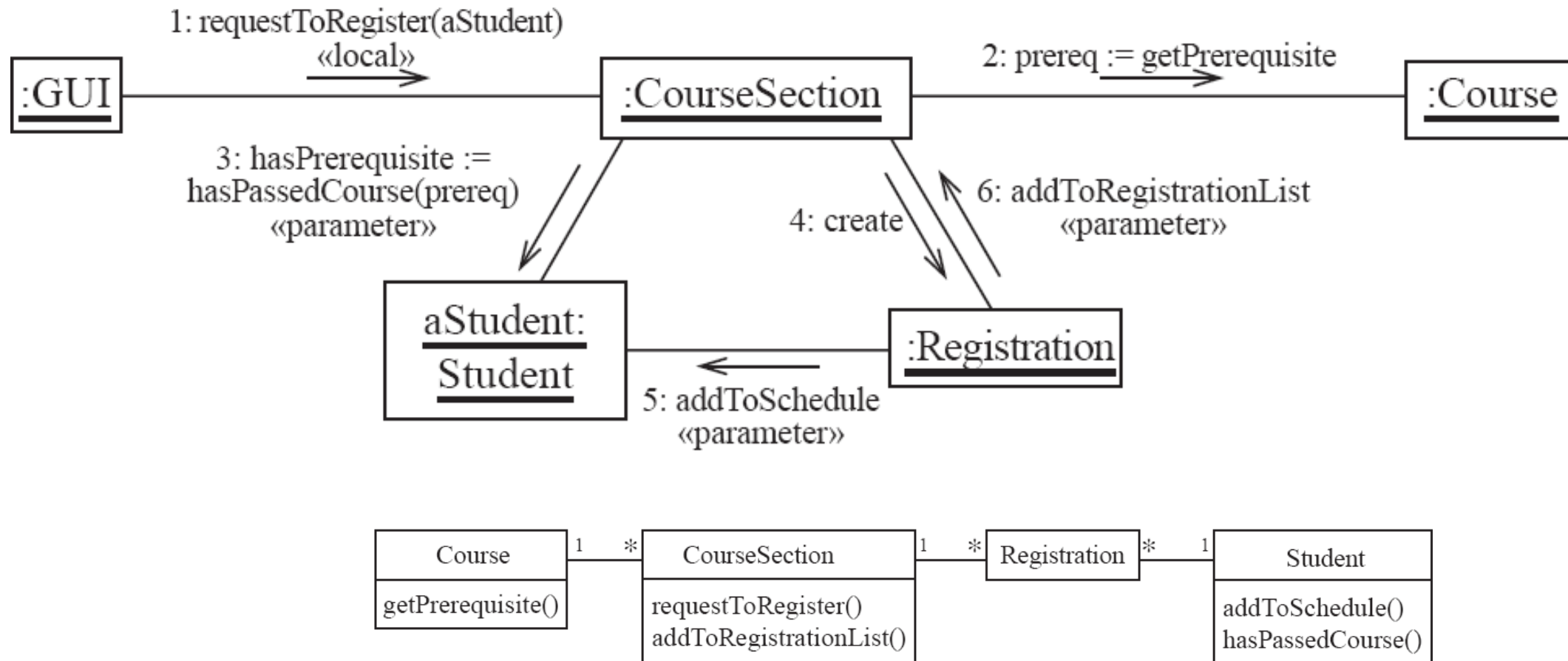
UML communication diagram

Example 3:



UML communication diagram

Example 3:



Sequence Diagram versus Communication Diagram

- Either a sequence diagram or a communication diagram can be used to depict the object interaction and sequence of messages passed among objects.
- the sequence diagram clearly shows the order in which messages are passed between objects, but seeing how the objects are connected to each other is more difficult.
- The communication diagram shows the layout of the objects, particularly how the objects are connected to each other.
- The message sequence is shown on both diagrams. Because the message sequence depicted on the communication diagram is less readily visible than on the sequence diagram.

يمكن استخدام **مخطط التسلسل** أو **مخطط الاتصال** لتمثيل تفاعل الكائنات وتسلسل الرسائل المتبادلة بينها.

مخطط التسلسل Sequence Diagram

يُظهر بوضوح ترتيب الرسائل التي يتم تمريرها بين الكائنات. لكن من الصعب رؤية كيفية ارتباط الكائنات ببعضها البعض من خلال هذا المخطط.

مخطط الاتصال Communication Diagram

يُظهر **تخطيط الكائنات** بشكل أوضح، وخاصةً كيفية ارتباطها ببعضها البعض.

يُوضح أيضًا تسلسل الرسائل، لكن رؤية التسلسل الزمني للرسائل أقل وضوحًا مقارنةً بمخطط التسلسل.



Sequence Diagram versus Communication Diagram

مخطط التسلسل Sequence Diagram

- Sequence diagrams Make explicit the time ordering of the interaction.
 - Use cases make time ordering explicit too
 - So sequence diagrams are a natural choice when you build an interaction model from a use case.

- Make it easy to add details to messages, communication diagrams have less space for this
- Communication diagrams Can be seen as a projection of the class diagram

Might be preferred when you are *deriving* an interaction diagram from a class diagram.

Are also useful for *validating* class diagrams.

- يُظهر بشكل صريح الترتيب الزمني للتفاعل بين الكائنات.
- بما أن حالات الاستخدام Use Cases تُظهر أيضًا الترتيب الزمني، فإن مخطط التسلسل يُعتبر خيارًا طبيعيًا عند بناء نموذج التفاعل انطلاقًا من حالة استخدام.
- يُسهل إضافة تفاصيل دقيقة إلى الرسائل (مثل المعاملات والقيم المرجعة).
- بالمقابل، مخطط الاتصال يوفر مساحة أقل لتوضيح هذه التفاصيل.

◆ مخطط الاتصال Communication Diagram

- يمكن النظر إليه على أنه إسقاط لمخطط الفئات Class Diagram ، حيث يُظهر الكائنات وروابطها.
- يُفضل استخدامه عند اشتقاق مخطط التفاعل من مخطط الفئات.
- يُعتبر أيضًا أداة مفيدة لـ التحقق من صحة مخططات الفئات، لأنه يُظهر كيفية تفاعل الكائنات الممثلة في مخطط الفئات مع بعضها البعض.



Use case name: Make Order Request

Summary: Customer enters an order request to purchase items from the online shopping system. The customer's credit card is checked for validity and sufficient credit to pay for the requested catalog items.

Actor: Customer

Precondition: Customer has selected one or more catalog items.

Main sequence:

1. **Customer** provides order request and customer account Id to pay for purchase.
2. System retrieves **customer account information**, including the customer's credit card details.
3. System checks the customer's **credit card** for the purchase amount and, if approved, creates a credit card purchase authorization number.
4. System creates a **delivery order** containing order details, customer Id, and credit card authorization number.
5. System confirms approval of purchase and displays order information to customer.
6. System sends **email** confirmation to customer.

Make Order Request Example

حالة استخدام: تقديم طلب شراء Make Order Request

الملخص Summary

يدخل العميل طلب شراء لاقتناء عناصر من نظام التسوق الإلكتروني. يقوم النظام بالتحقق من صلاحية بطاقة الائتمان الخاصة بالعميل والتأكد من توفر رصيد كافٍ لدفع ثمن العناصر المطلوبة من الكتالوج.

الممثل Actor

• العميل Customer

الشرط المسبق Precondition

- يجب أن يكون العميل قد اختار عنصرًا واحدًا أو أكثر من الكتالوج.

التسلسل الرئيسي Main Sequence

1. يقوم العميل بتقديم طلب الشراء مع رقم حسابه لدفع قيمة المشتريات.
2. يسترجع النظام معلومات حساب العميل، بما في ذلك تفاصيل بطاقة الائتمان.
3. يتحقق النظام من بطاقة الائتمان الخاصة بالعميل بالنسبة لقيمة المشتريات، وإذا تمت الموافقة، ينشئ رقم تفويض شراء بالبطاقة.
4. ينشئ النظام أمر تسليم يحتوي على تفاصيل الطلب، رقم العميل، ورقم تفويض البطاقة.
5. يؤكد النظام الموافقة على عملية الشراء ويعرض معلومات الطلب للعميل.
6. يرسل النظام رسالة بريد إلكتروني لتأكيد الطلب إلى العميل.



Make Order Request Example



Alternative sequences:

Step 2: If customer does not have account, the system prompts the customer to provide information in order to create a new account. The customer can either enter the account information or cancel the order.

Step 3: If authorization of the customer's credit card is denied (e.g., invalid credit card or insufficient funds in the customer's credit card account), the system prompts the customer to enter a different credit card number. The customer can either enter a different credit card number or cancel the order.

Postcondition: System has created a delivery order for the customer.

التسلسلات البديلة Alternative Sequences

- **الخطوة 2:** إذا لم يكن لدى العميل حساب، يطلب النظام من العميل إدخال معلومات لإنشاء حساب جديد. يمكن للعميل إدخال المعلومات أو إلغاء الطلب.
- **الخطوة 3:** إذا تم رفض تفويض بطاقة الائتمان (مثل بطاقة غير صالحة أو رصيد غير كافٍ)، يطلب النظام من العميل إدخال رقم بطاقة آخر. يمكن للعميل إدخال بطاقة جديدة أو إلغاء الطلب.

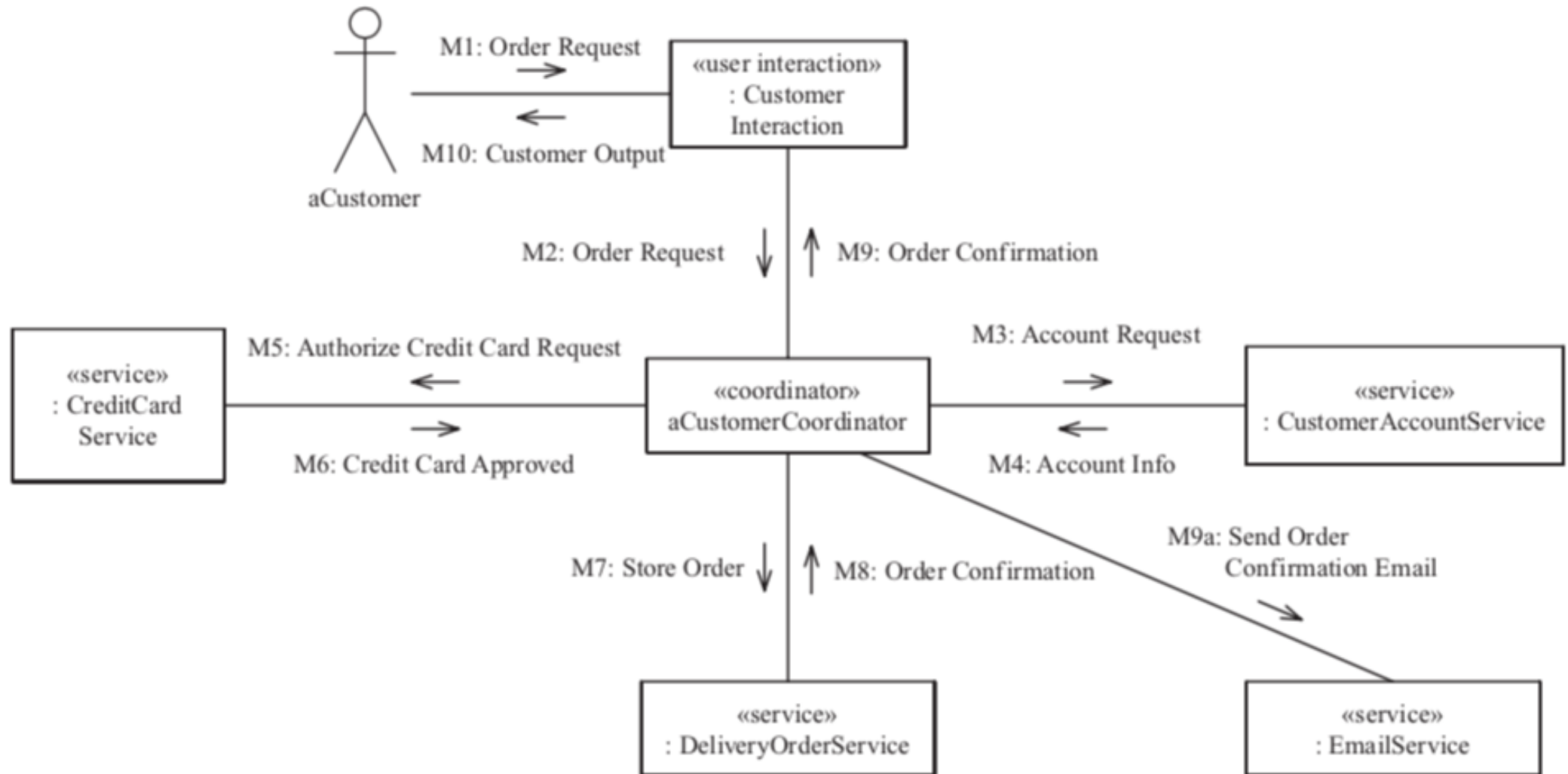
الشرط اللاحق Postcondition

- يكون النظام قد أنشأ أمر تسليم للعميل.



Make Order Request Example

Communication diagram for the Make Order Request use case: main sequence



Communication diagram for the Make Order Request use case: main sequence

The message descriptions are as follows:

M1: The customer provides order request to Customer Interaction.

M2: Customer Interaction sends the order request to Customer Coordinator.

M3, M4: Customer Coordinator sends the account request to Customer Account Service and receives the account information, including the customer' s credit card details.

M5: Customer Coordinator sends the customer' s credit card information to Credit Card Service.

M6: Credit Card Service sends a credit card approval to Customer Coordinator.

M7, M8: Customer Coordinator sends order request to Delivery Order Service.

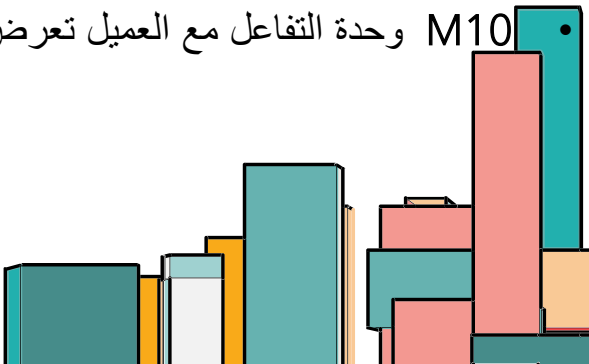
M9, M9a: Customer Coordinator sends the order confirmation to Customer Interaction and sends an email of the order confirmation to the customer via the Email Service.

M10: Customer Interaction outputs the order confirmation to the customer.

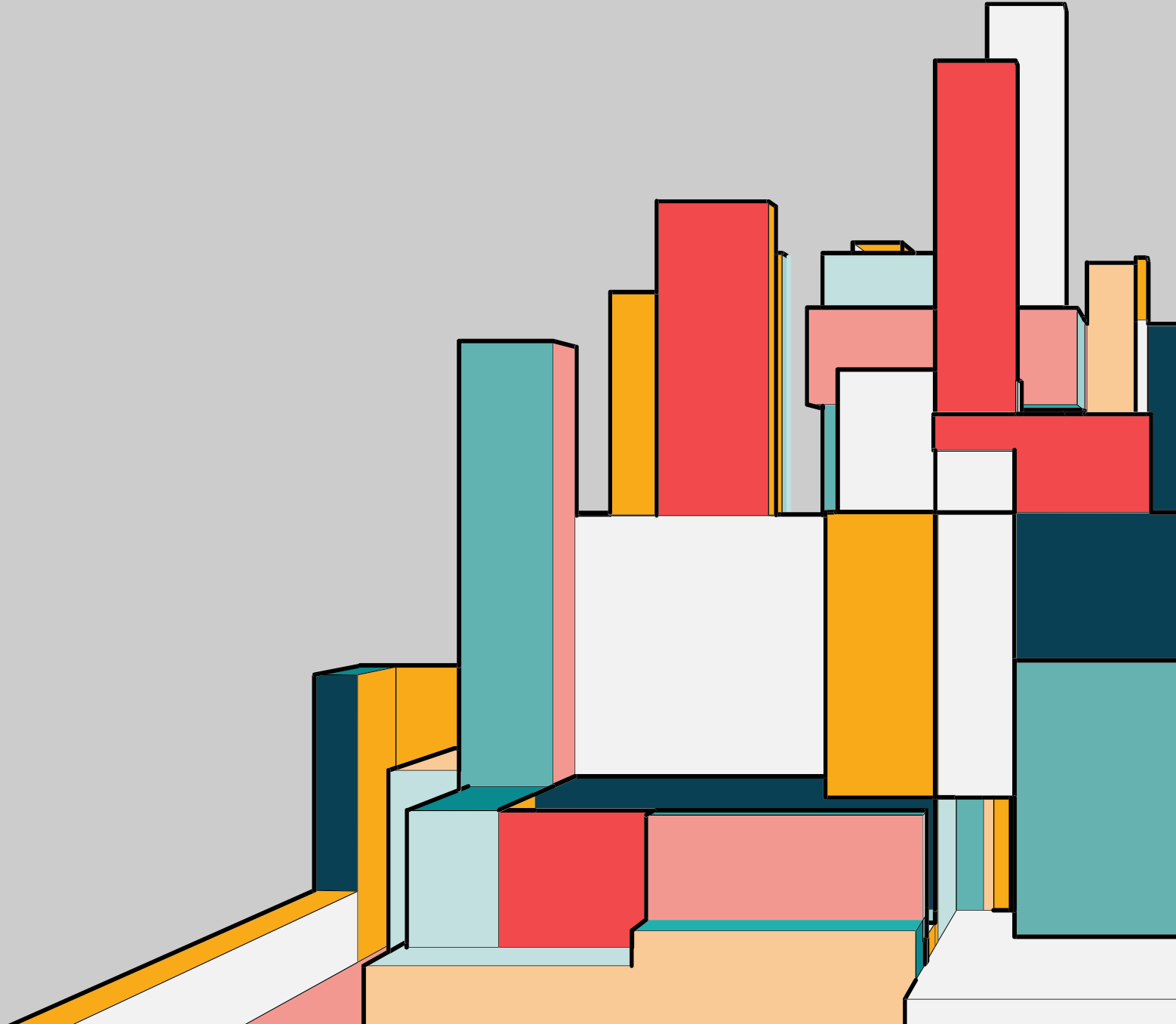
Exercise: Make Order Request

الرسائل الأساسية

- M1 العميل يقدم طلب الشراء إلى وحدة التفاعل مع العميل Customer Interaction
- M2 وحدة التفاعل مع العميل ترسل طلب الشراء إلى منسق العميل Customer Coordinator
- M3, M4 منسق العميل يرسل طلب الحساب إلى خدمة حساب العميل Customer Account Service ويتلقى معلومات الحساب، بما في ذلك تفاصيل بطاقة الائتمان.
- M5 منسق العميل يرسل معلومات بطاقة الائتمان إلى خدمة بطاقة الائتمان Credit Card Service
- M6 خدمة بطاقة الائتمان ترسل الموافقة على البطاقة إلى منسق العميل.
- M7, M8 منسق العميل يرسل طلب الشراء إلى خدمة أوامر التسليم Delivery Order Service
- M9, M9a منسق العميل يرسل تأكيد الطلب إلى وحدة التفاعل مع العميل، ويرسل أيضًا بريدًا إلكترونيًا لتأكيد الطلب إلى العميل عبر خدمة البريد الإلكتروني Email Service
- M10 وحدة التفاعل مع العميل تعرض تأكيد الطلب للعميل.



APPENDIXES



ASSIGNMENT 1

- The previous questions helps in **defining the scope** of that project. Answers helps in having a clear consensus about if this production project could be conducted or not.
- If there some technical, legal, expertise- related, political, or other could arise, and couldn't be mitigated, then the production project is not feasible.
- If the dedicated budget was not reasonable at all then, Boeing will notify Libyan Airline that this plane couldn't be produced and delivered to Libya.
- In contrary, if the project shows feasibility then, the rest of information will help in defining the scope of the project.