

၂၀၁၀ နေ့ပါမ်မာ

- Close book

Unified Modeling Language

Parinya Ekparinya

Parinya.Ek@kmitl.ac.th

Outline

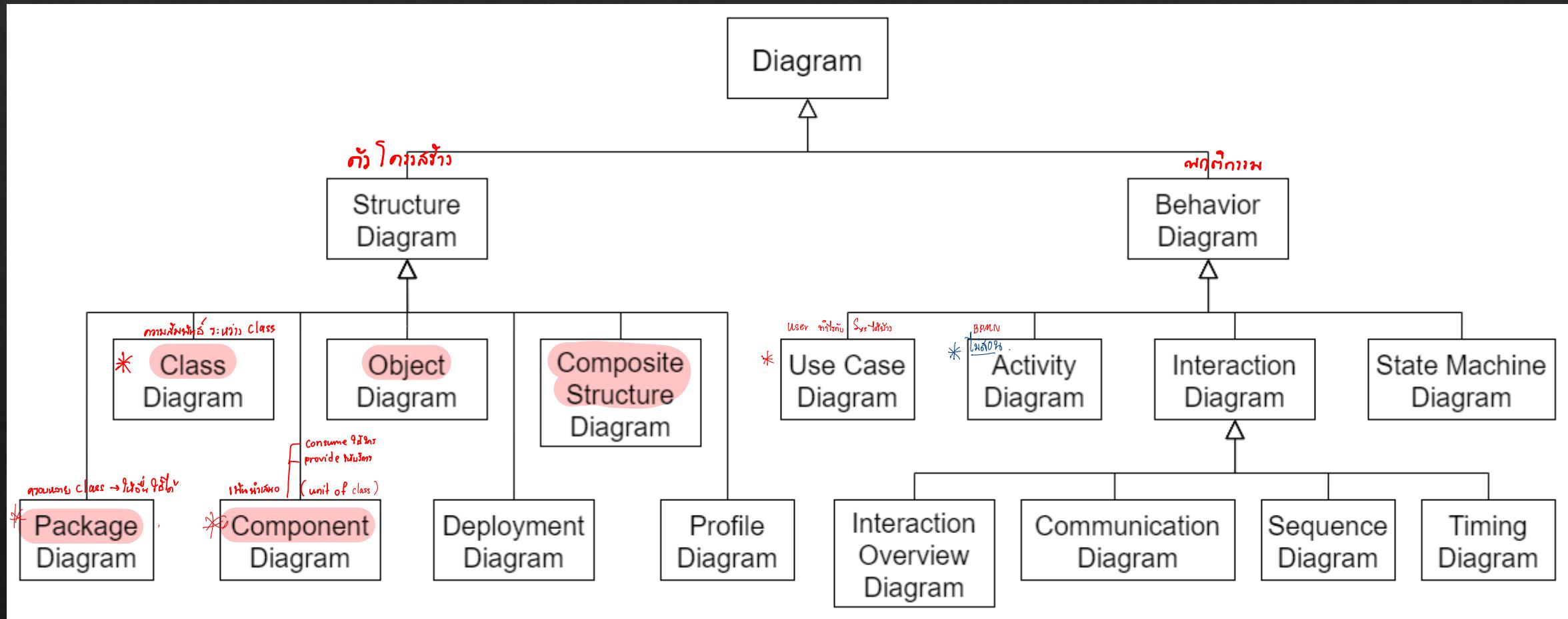
- ❖ Unified Modeling Language (UML)
- ❖ Class Diagram
- ❖ Activity Diagram
- ❖ Component Diagram
- ❖ Deployment Diagram

Unified Modeling Language (UML)

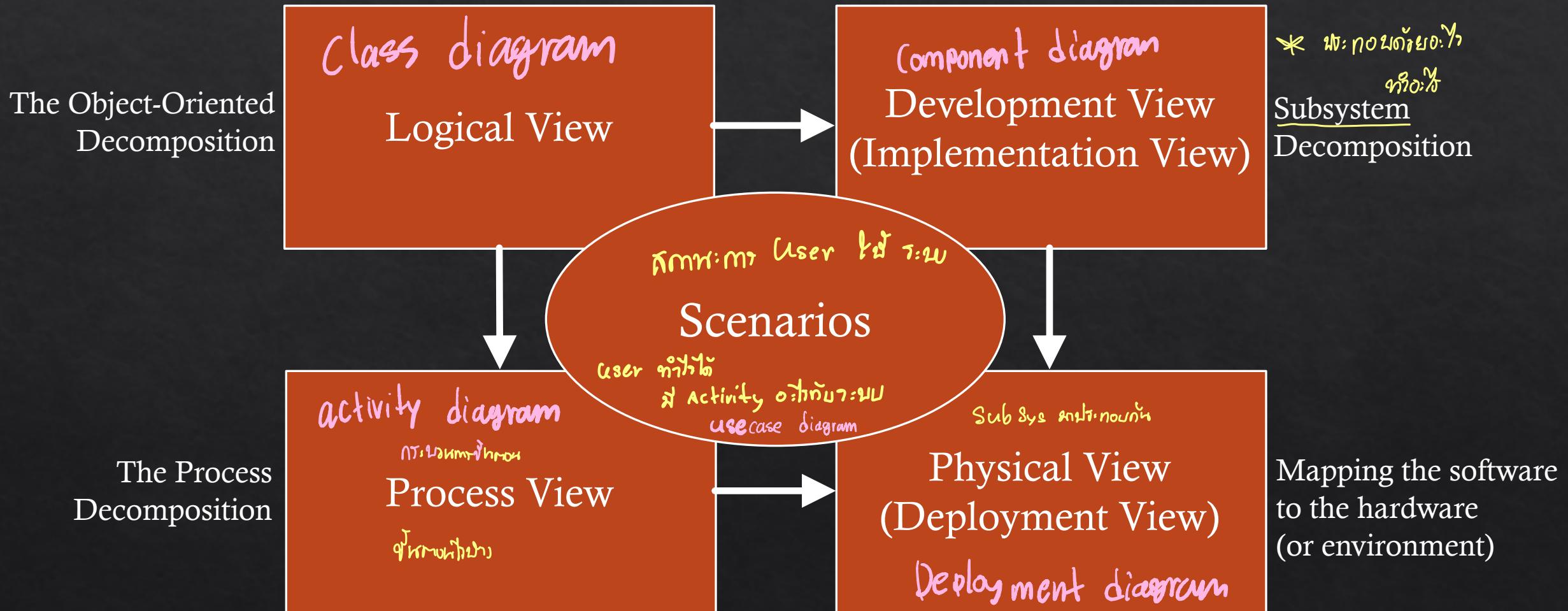
- ❖ The UML is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system.
- ❖ The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components.
- ❖ The UML was developed at Rational Software in 1994–1995.
- ❖ The UML was also published by the ISO as an approved ISO standard in 2005.
- ❖ The latest version is 2.5.1, which is published in 2017.
For specification: <https://www.omg.org/spec/UML/2.5.1/About-UML/>

↗ ภูมิทัศน์ prototype
 * OOP ภูมิทัศน์ class
 ความ + ผู้ใช้
 OOP Class = state + Behavior

UML Diagram Types



4+1 View Model of Software Architecture



Kruchten, P. B. (1995). The 4+1 view model of architecture. *IEEE software*, 12(6), 42-50.

Logical View

អាមេរិក —> ស្ថាប័នបំណុល ស៊ីហ៍ការកំរា

- ❖ Logical view describes the abstract descriptions of a system's parts.
 - ❖ This view is used to model what a system is made up of and how the parts interact with each other.
 - ❖ The system is decomposed into a set of key abstractions in the form of objects or object classes. They exploit the principles of abstraction, encapsulation, and inheritance.
 - ❖ With reference to UML, logical view typically contains class diagrams.

Process View

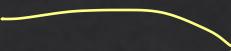
ឯងជាយ

ទីន

- ❖ Process view describes the processes within your system.
ឯងការគាំទ្រ និងការគាំទ្រ ដែលមានការងារ ឬការសម្រេច
- ❖ A *process* is a grouping of tasks that form an executable unit.
- ❖ The software is partitioned into a set of independent tasks. A task is a separate thread of control, that can be scheduled individually on one processing node.
- ❖ Several styles would fit the process view, pipes and filters for example.
ពីអតិថជ្ជរដ្ឋាភិបាល
present យុទ្ធសាស្ត្រ
- ❖ With reference to UML, process view typically contains activity diagrams.

Development View

ດៃខិសក្រណីអង្គ



- ❖ Development view describes how your system's parts are organized into modules and components.
- ❖ Typically, this view focuses on the configuration management of a system; what components depend on what. (មិនកំពុង ចិត្ត , ចាប់ផ្តើម service C_B .
- ❖ The development view is also known as **implementation view**.
- ❖ A layered style would fit the process view. Each layer has a well-defined responsibility.
- ❖ With reference to UML, development view typically contains **component diagrams**.

Arch អេរ៉ូក្រា

Physical View

- ❖ Physical view describes how the system's components, as described in the other views, are arranged in their environments. ໂດຍມີຄວາມ
↗ Ex library , execute
- ❖ The diagrams in this view show how the abstract parts map into the final deployed system.
- ❖ Thus, the physical view is also known as deployment view.
- ❖ With reference to UML, physical view typically contains deployment diagrams.

Scenarios

Exter

នគរបាល នឹង activity ខែក្រោម ផ្តល់ព័ត៌មាន

- ❖ Scenarios describe the functionality of the system being modeled from the perspective of its users and/or the outside world.
ប្រព័ន្ធសម្រាប់ប្រើប្រាស់
- ❖ The scenarios are in some sense an abstraction of the most important requirements.
- ❖ The scenarios are needed to describe what the system is supposed to do.
- ❖ This view is also known as use case view.
- ❖ This view is redundant with the other ones (hence the “+1”).
- ❖ With reference to UML, scenarios are typically expressed through use case diagrams.

Learning UML

- ❖ <https://www.omg.org/spec/UML/2.5/About-UML/>
- ❖ <https://www.uml-diagrams.org/>

S...S

+ public do : Attributed class
- private do

+ method (doInstr : type) : searchResult
- Method () : searchResult ...> return type
static method

* static នៃជូន class

ទី៣ Class Camel

} Attribute

} method

Visibility រូបចារា

- private → subclass (អង់គ្គល់)

+ public តិចកំណើន

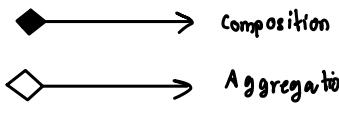
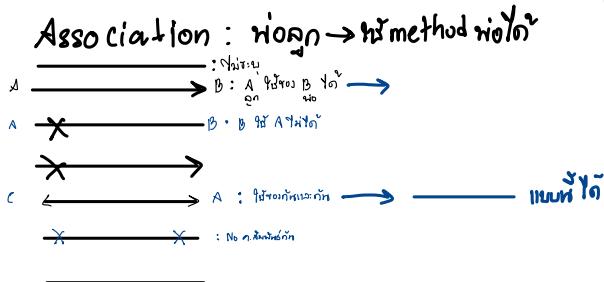
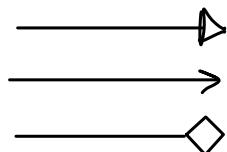
protected → subclass តុល់

~ package → default

Abstract class ຕົວຢ່ານ

LL Interface class ...> default public ເລີມວ
ນີ້ class

၁. ສິນພິບອົງຮວງ class



Composite / Composition : ສ່ານຫົວ (Object-oriented) ຢົດ Class ສົ່ວໂລມ
ໃນຂະໜາດຂອງການສ່ານຫົວ



share / Aggregation :

ໃຫຍ່ກຳ

95
A ←
ມີນ method static ອີ class B

Dependency

→ ທີ່ object ເກີນໄວ້

→ UML ຮຶບຮັກ

→ Implement Interface

→ Inheritance / Generalization



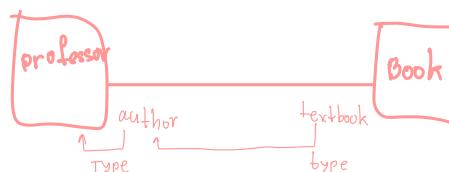
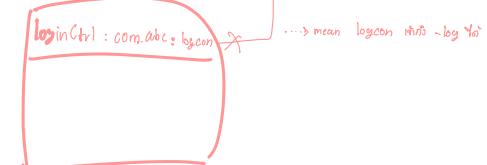
Multiplicity

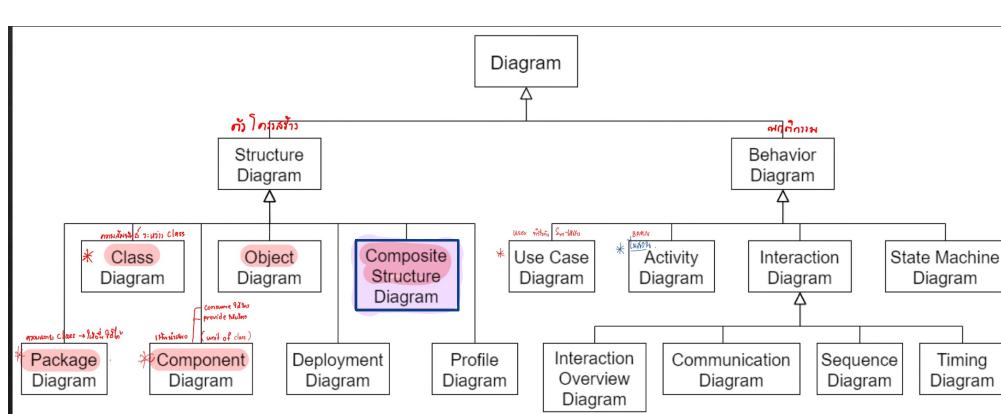
ນີ້ໂດຍຕ້ອງນີ້ມາດີນີ້

min	max	max
0..1	*	1
1..*		*
0..*		0



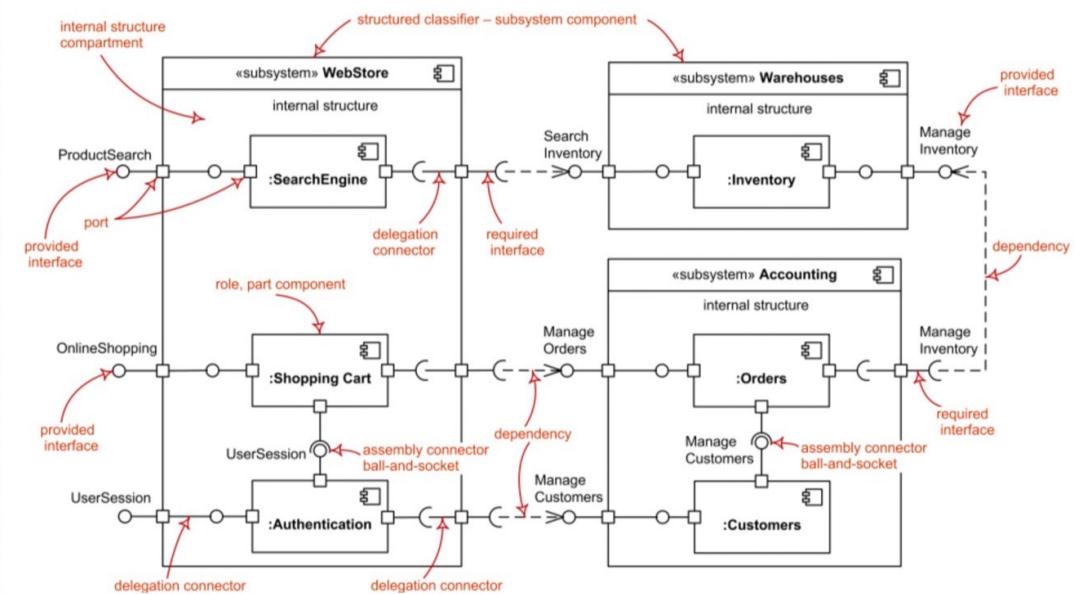
~log (private)





Show Components
in terms of SOA

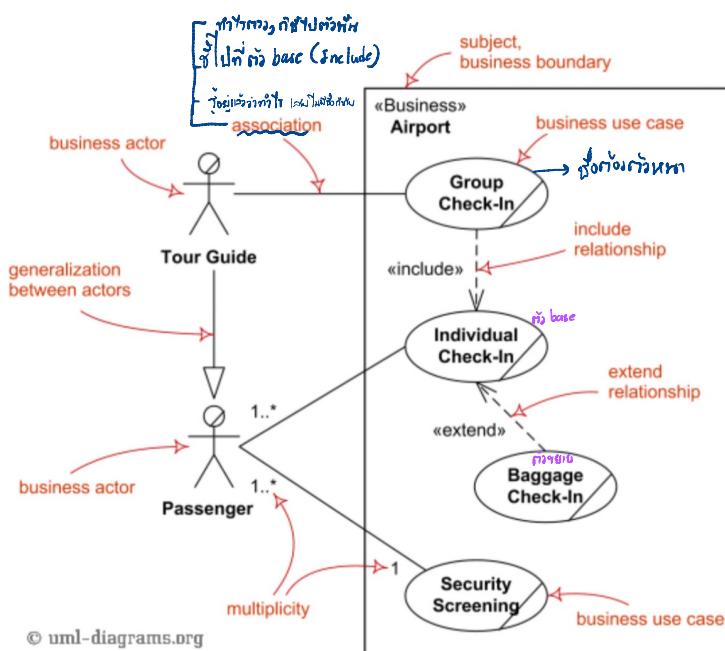
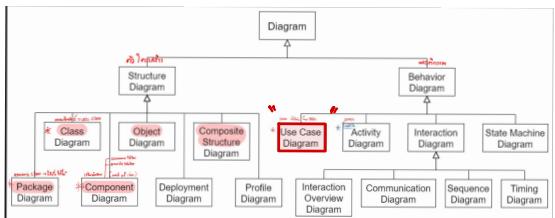
- C.: Components
 - ផ្តល់ ទទួល
 - ពិនិត្យ និរនោគភាព
 - port
 - relationship links: Components
- Logical C.
- physical C. (EJB c., COM+.NET component)



The major elements of UML component diagram - component, provided interface, required interface, port, connectors.

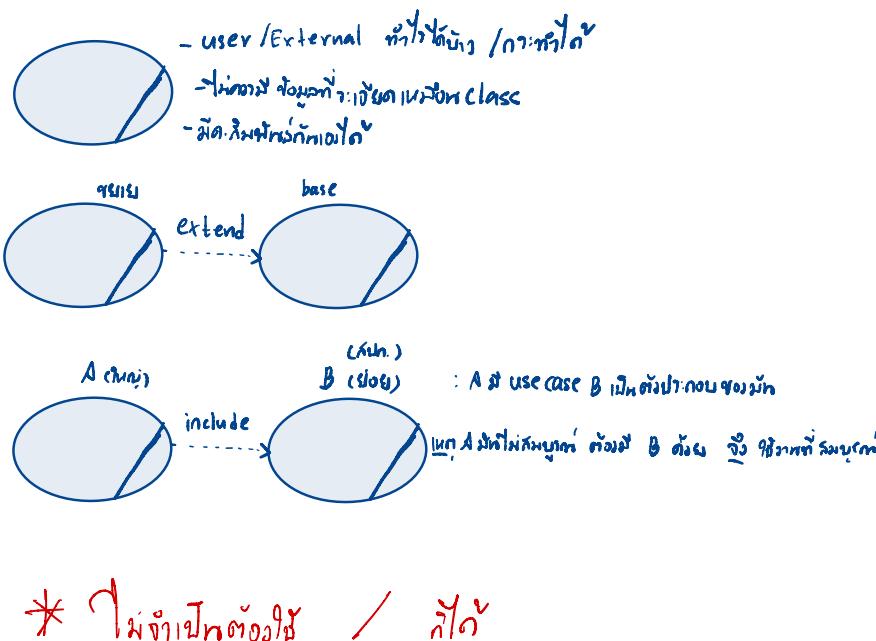
Use case

ມີສັບ ຂຶ້ນຂອງ External ມີຫົວໜ້າ: ນຸ່ງດັກ



© uml-diagrams.org

Major elements of business use case diagram - **business actor**, **business use case**, **business boundary**, **include** and **extend** relationships.



* ຜົນກົດເມືອງ / ຖະໜາ