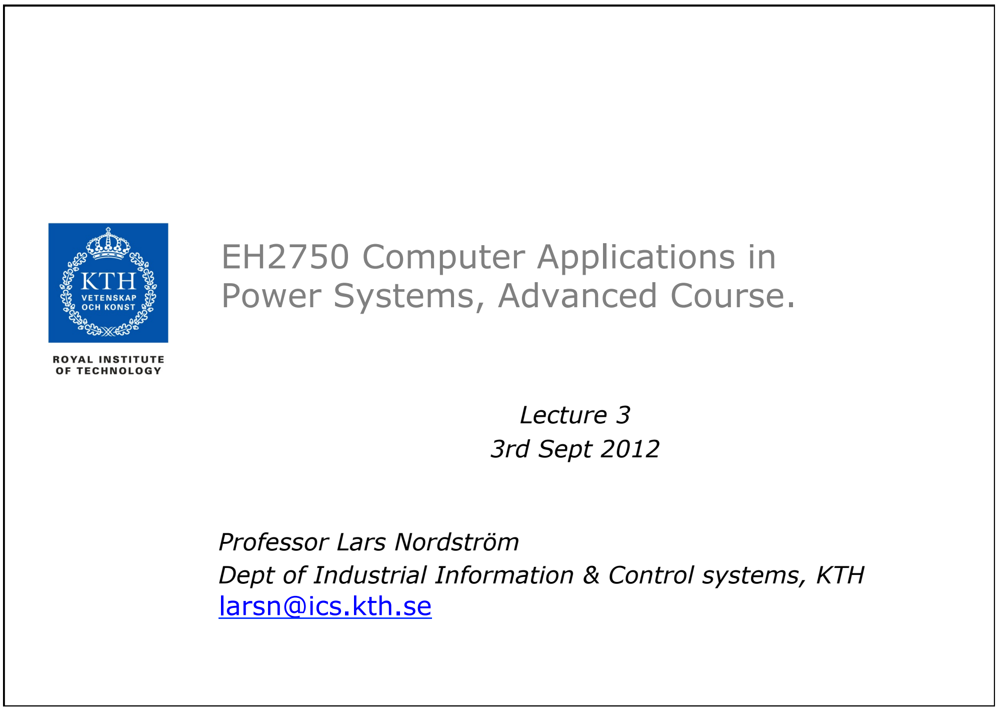
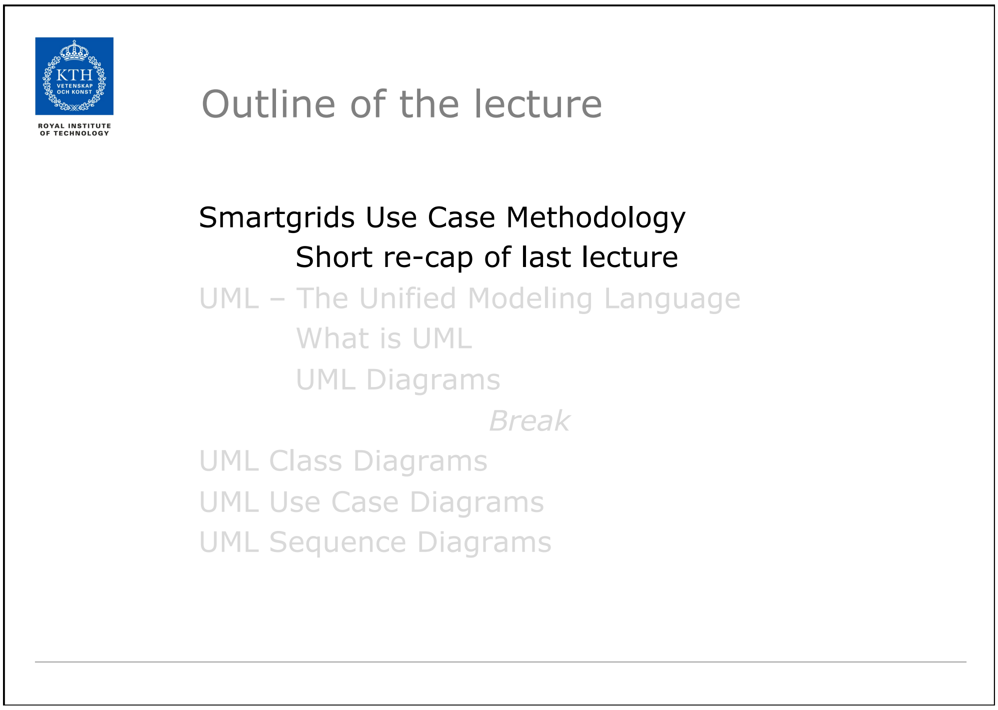


EH2750 Computer Applications in Power Systems, Advanced Course.

Lecture 3
3rd Sept 2012

Professor Lars Nordström
Dept of Industrial Information & Control systems, KTH
larsn@ics.kth.se



Outline of the lecture

Smartgrids Use Case Methodology
- Short re-cap of last lecture

UML – The Unified Modeling Language
- What is UML
- UML Diagrams

Break

UML Class Diagrams
UML Use Case Diagrams
UML Sequence Diagrams

A structured approach is needed

- The IEC Publically Available Specification 62559
  - Developed initially by the EPRI in the US as part of the Intelligrid project
  - Adopted by the IEC as a PAS in 2008
- The 62559 is not a standard, it is instead a suggested way to work with developing requirements on new computer applications for power systems.

IEC/PAS 62559

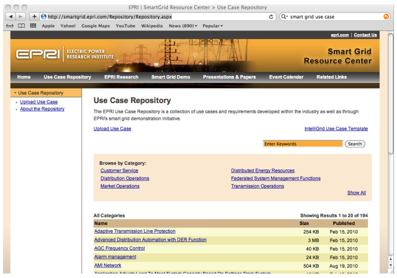
Edition 1.0 2008-01

PUBLICLY AVAILABLE SPECIFICATION

IntelliGrid Methodology for Developing Requirements for Energy Systems

Review existing Use Cases

- The EPRI Use Case Repository
  - A huge collection of Use Cases exist online
  - Developed initially in the Intelligrid project in the late 1990s and continued during 2008+
  - The level of granularity & completeness very varied



Use Case Repository

The EPRI Use Case Repository is a collection of use cases and requirements developed within the industry as well as through EPRI's smart grid demonstration initiative.

Upload Use Case

IntelliGrid Use Case Template

Browse by Category:

Customer Service

Distribution Operations

Market Operations

Distributed Energy Resources

Federated System Management Functions

Transmission Operations

Show All

All Categories

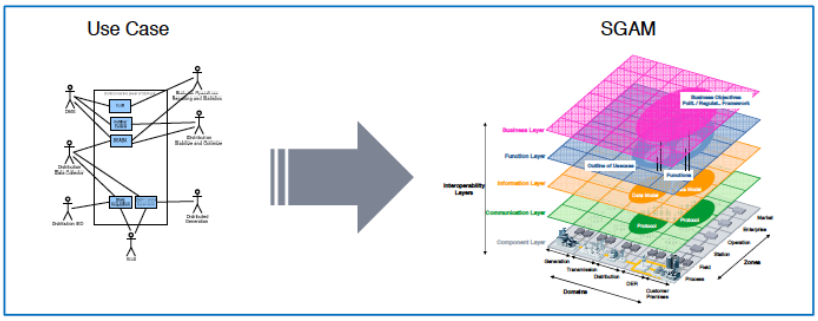
Showing Results 1 to 20 of 194

Name	Size	Published
Adaptive Transmission Line Protection	254 KB	Feb 15, 2010
Advanced Distribution Automation with DER Function	3 MB	Feb 15, 2010
AGC Frequency Control	40 KB	Feb 15, 2010
Alarm management	24 KB	Feb 15, 2010
AMI Network	504 KB	Aug 19, 2010

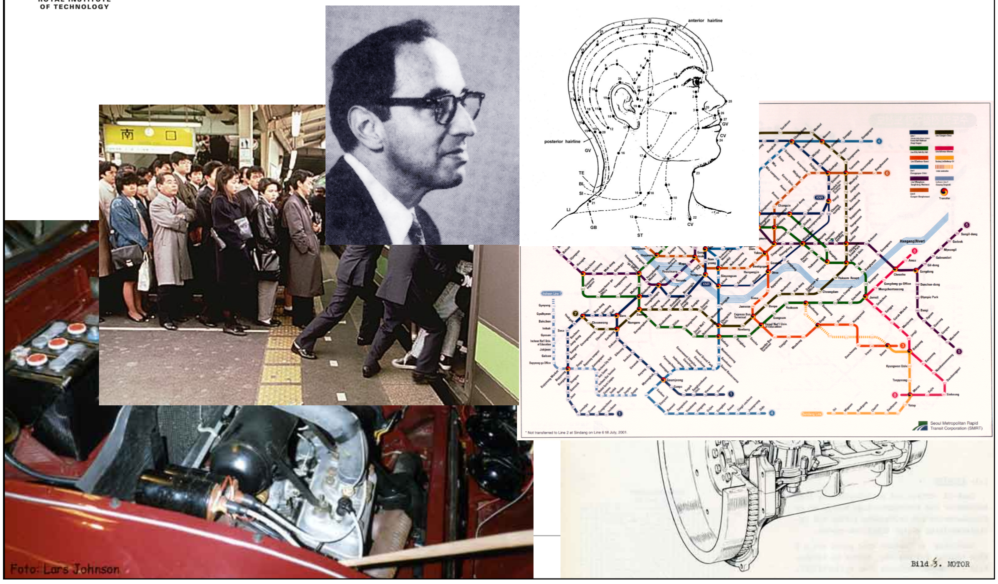
Step 4 – Drafting Use Cases

- Important contents of a Use Case
  - The **goal** of the use case, which is usually its name. e.g. "Utility remotely connects or disconnects customer".
  - The **narrative**. A short English text version of the story.
  - The **actors**. An actor is anything in the system that communicates. e.g. a "customer" or a "meter".
  - The **assumptions** that the use case is based on. These can constitute requirements in and of themselves.
  - The **contracts** and **preconditions** that exist between the actors,
  - The **triggering** event that led to the scenario taking place.
  - The **steps**. A numbered list of events that tell the story in detail.

Mapping the Use Case to architecture

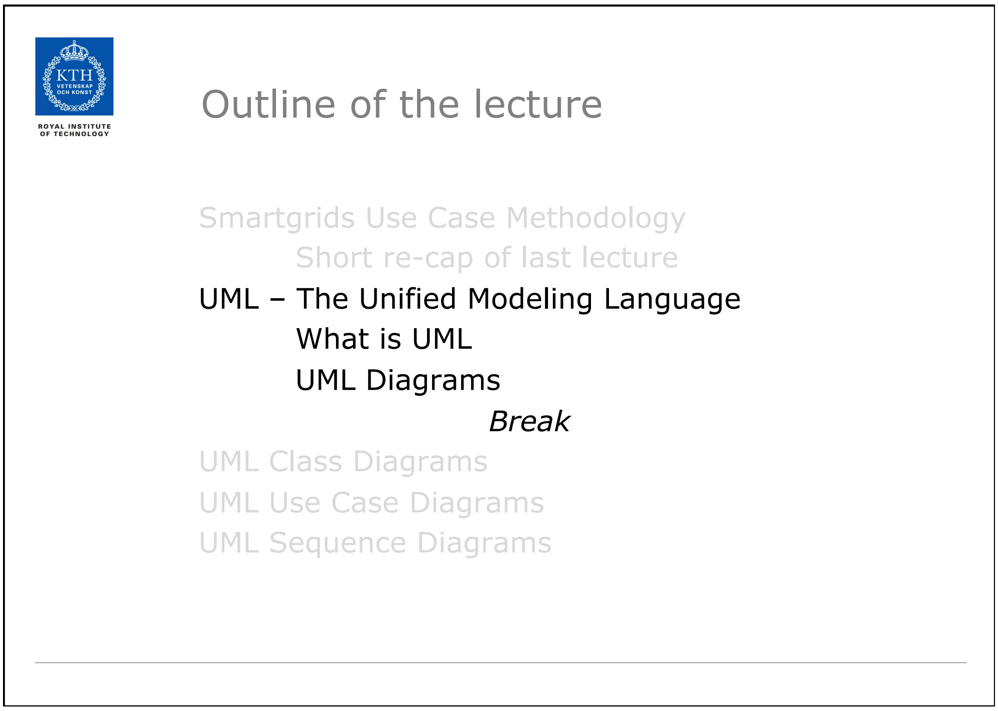


How to make things understandable



Step 4 – Drafting Use Cases

- Important contents of a Use Case
  - The **goal** of the use case, which is usually its name. e.g. "Utility remotely connects or disconnects customer".
  - The **narrative**. A short English text version of the story.
  - The **actors**. An actor is anything in the system that communicates. e.g. a "customer" or a "meter".
  - The **assumptions** that the use case is based on. These can constitute requirements in and of themselves.
  - The **contracts** and **preconditions** that exist between the actors,
  - The **triggering** event that led to the scenario taking place.
  - The **steps**. A numbered list of events that tell the story in detail.



Outline of the lecture

Smartgrids Use Case Methodology

Short re-cap of last lecture

UML – The Unified Modeling Language

What is UML

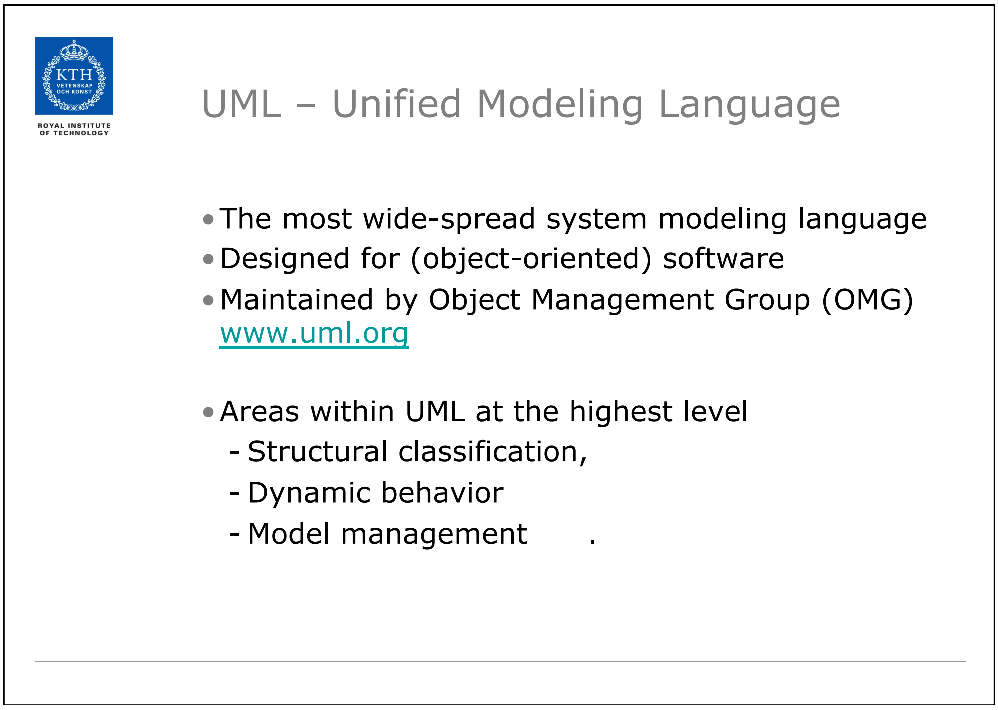
UML Diagrams

Break

UML Class Diagrams

UML Use Case Diagrams

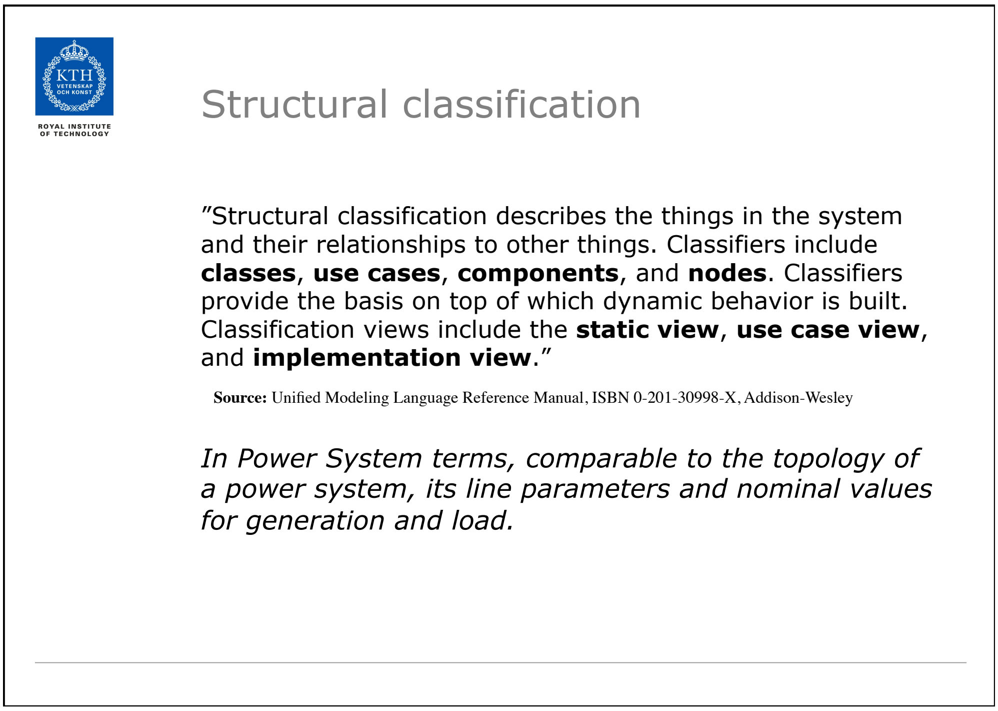
UML Sequence Diagrams



UML – Unified Modeling Language

- The most wide-spread system modeling language
- Designed for (object-oriented) software
- Maintained by Object Management Group (OMG) www.uml.org

- Areas within UML at the highest level
  - Structural classification,
  - Dynamic behavior
  - Model management .



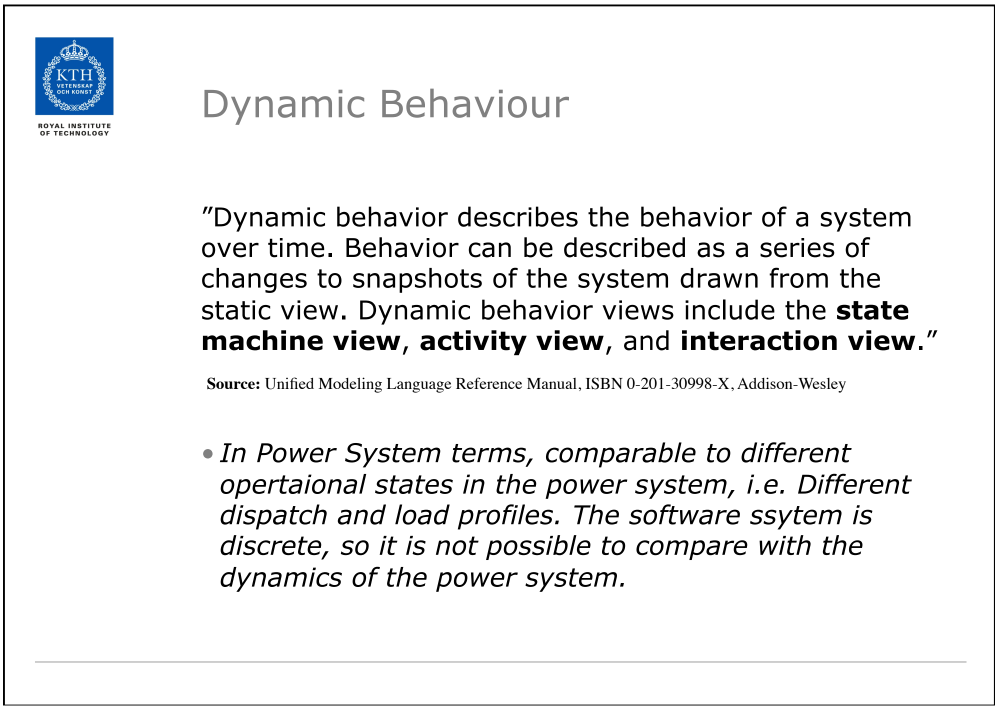
ROYAL INSTITUTE OF TECHNOLOGY

Structural classification

”Structural classification describes the things in the system and their relationships to other things. Classifiers include **classes**, **use cases**, **components**, and **nodes**. Classifiers provide the basis on top of which dynamic behavior is built. Classification views include the **static view**, **use case view**, and **implementation view**.”

Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

In Power System terms, comparable to the topology of a power system, its line parameters and nominal values for generation and load.



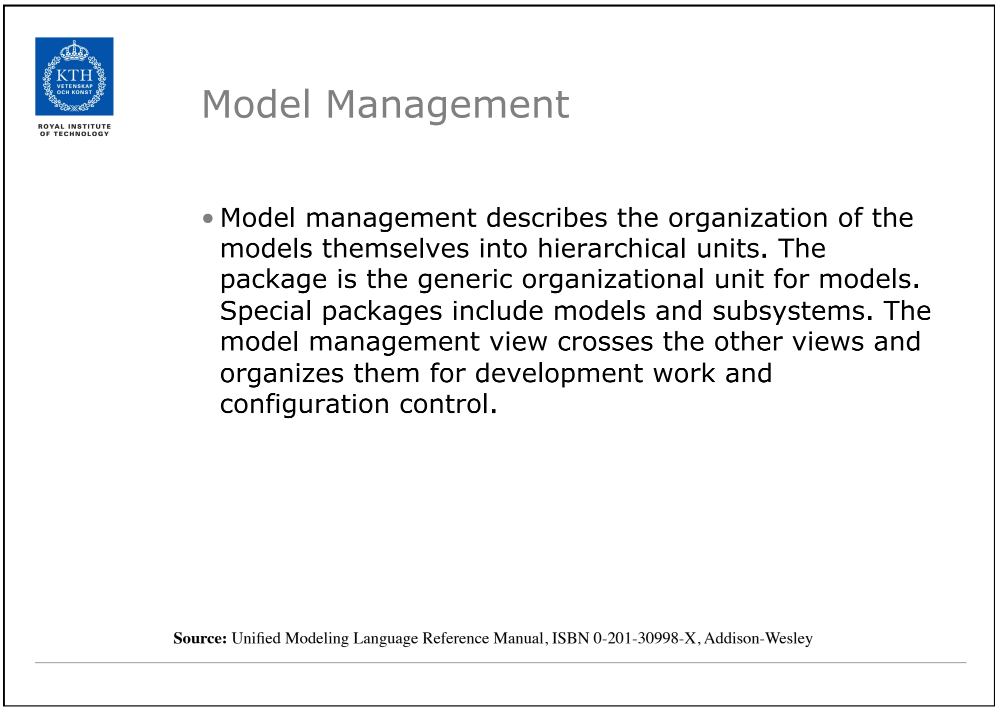
ROYAL INSTITUTE OF TECHNOLOGY

Dynamic Behaviour

”Dynamic behavior describes the behavior of a system over time. Behavior can be described as a series of changes to snapshots of the system drawn from the static view. Dynamic behavior views include the **state machine view**, **activity view**, and **interaction view**.”

Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

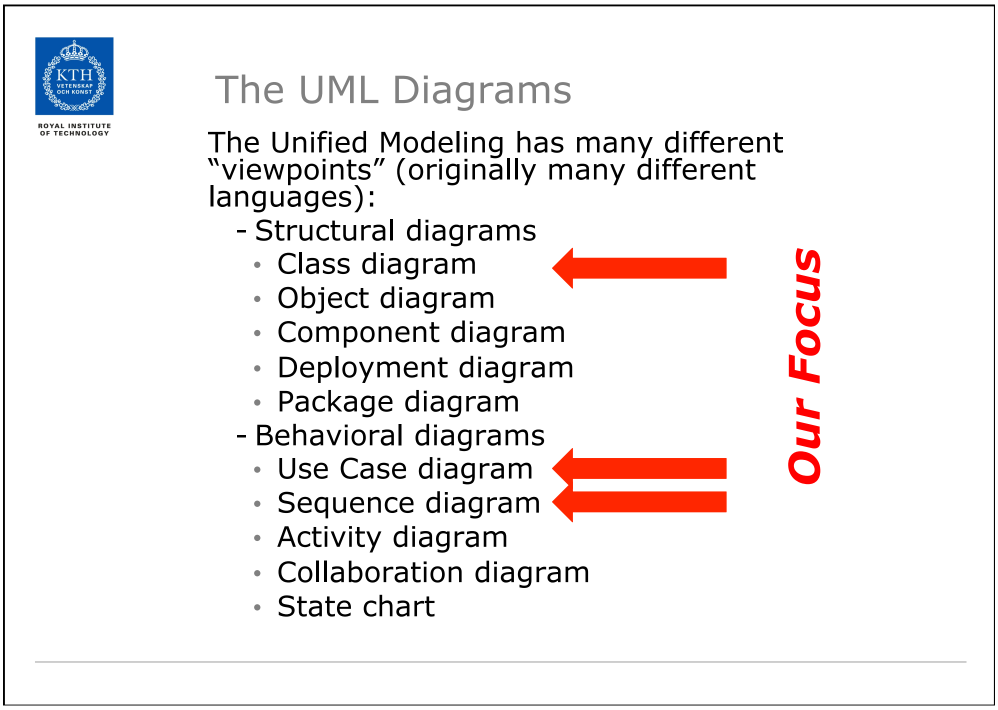
- *In Power System terms, comparable to different opertaional states in the power system, i.e. Different dispatch and load profiles. The software ssytem is discrete, so it is not possible to compare with the dynamics of the power system.*



Model Management

- Model management describes the organization of the models themselves into hierarchical units. The package is the generic organizational unit for models. Special packages include models and subsystems. The model management view crosses the other views and organizes them for development work and configuration control.

Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley



The UML Diagrams

The Unified Modeling has many different "viewpoints" (originally many different languages):

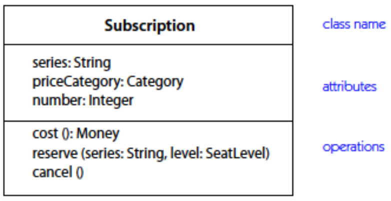
- Structural diagrams
  - Class diagram
  - Object diagram
  - Component diagram
  - Deployment diagram
  - Package diagram
- Behavioral diagrams
  - Use Case diagram
  - Sequence diagram
  - Activity diagram
  - Collaboration diagram
  - State chart

Our Focus

ROYAL INSTITUTE OF TECHNOLOGY

Class diagrams

- A class represents a discrete concept within the application being modeled—a physical thing (such as an airplane), a business thing (such as an order), a logical thing (such as a broadcasting schedule), an application thing (such as a cancel button), a computer thing (such as a hash table), or a behavioral thing (such as a task).



- An object is an instance of a class. I.e. it has an identity and a specific set of values.

Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

ROYAL INSTITUTE OF TECHNOLOGY

Associations between classes

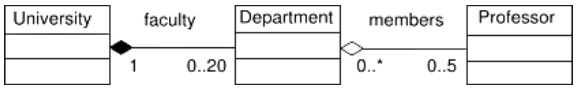
Table 4-2: *Kinds of Relationships*

Relationship	*Function*	*Notation*
association	A description of a connection among instances of classes	——
dependency	A relationship between two model elements	- - - ->
flow	A relationship between two versions of an object at successive times	- - - ->
generalization	A relationship between a more general description and a more specific variety of the general thing, used for inheritance	——▷
realization	Relationship between a specification and its implementation	- - - -▷
usage	A situation in which one element requires another for its correct functioning	- - - ->

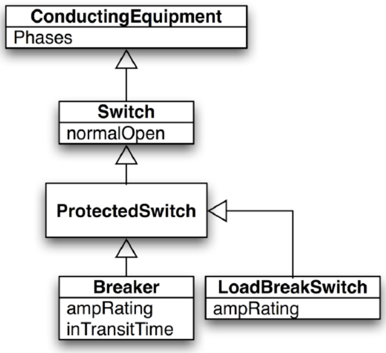
Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

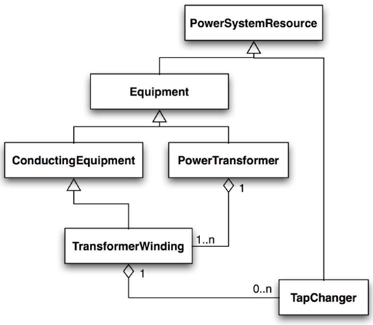
Two additional relations Aggregation & Composition

- A university consists of (**composition**) departments (e.g., computer science), and each department has (**aggregates**) a number of professors.
- If the university closes, the departments will no longer exist, but the professors in those departments will continue to exist.



Class Diagram – two examples





Source: An Introduction to IEC 61970-301 & 61968-11:The Common Information Model, A. McMorran, University of Strathclyde

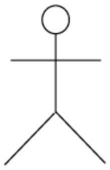
Use Case Diagram

”The use case view captures the behavior of a system, subsystem, or class as it appears to an outside user. It partitions the system functionality into transactions meaningful to actors—idealized users of a system”

Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

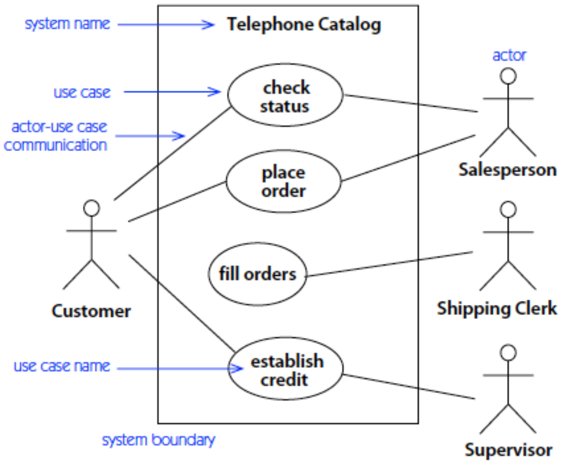
Actors in around the system

- An actor is an idealization of an external person, process, or thing interacting with a system, subsystem, or class.
- An actor may be a human, another computer system, or some executable process.
- Actors may be defined in generalization hierarchies, in which an abstract actor description is shared and augmented by one or more specific actor descriptions



Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

Use Case diagram



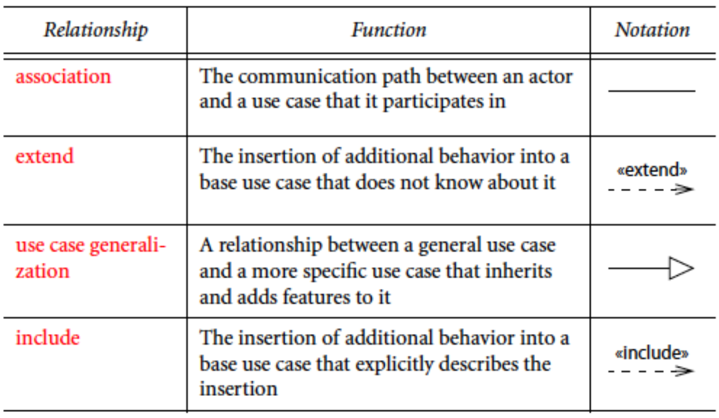
Naming confusion:

UML Use Case is a SGAM function

Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

Use Case relations

Table 5-1: *Kinds of Use Case Relationships*



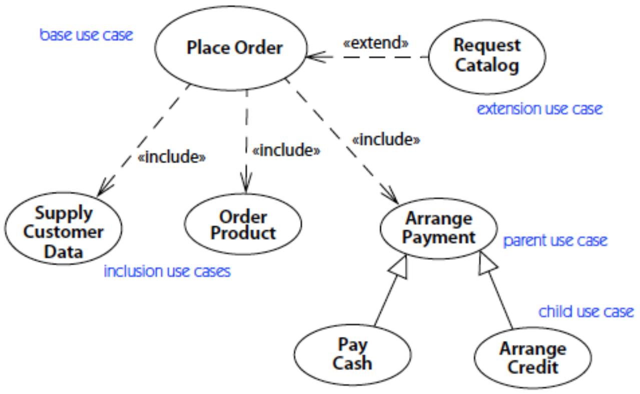
Relationship	*Function*	*Notation*
association	The communication path between an actor and a use case that it participates in	———
extend	The insertion of additional behavior into a base use case that does not know about it	«extend» - - - - ->
use case generalization	A relationship between a general use case and a more specific use case that inherits and adds features to it	——▷
include	The insertion of additional behavior into a base use case that explicitly describes the insertion	«include» - - - - ->

Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

Use Case examples.

Naming confusion:

UML Use Case is a SGAM function



Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

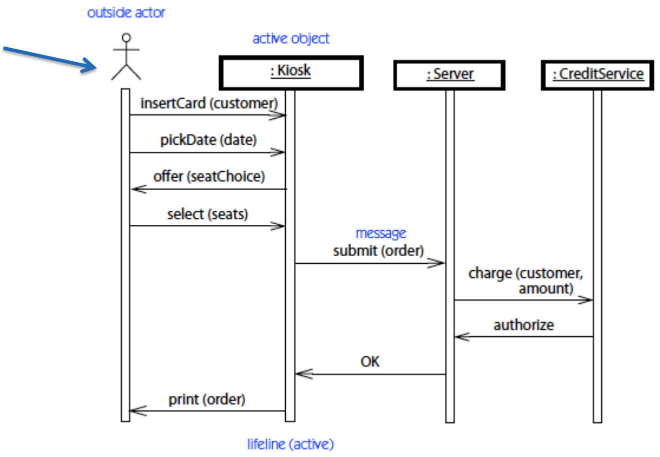
Sequence Diagrams

- A sequence diagram displays an interaction as a two-dimensional chart. The vertical dimension is the time axis; time proceeds down the page. The horizontal dimension shows the classifier roles that represent individual objects in the collaboration.

Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

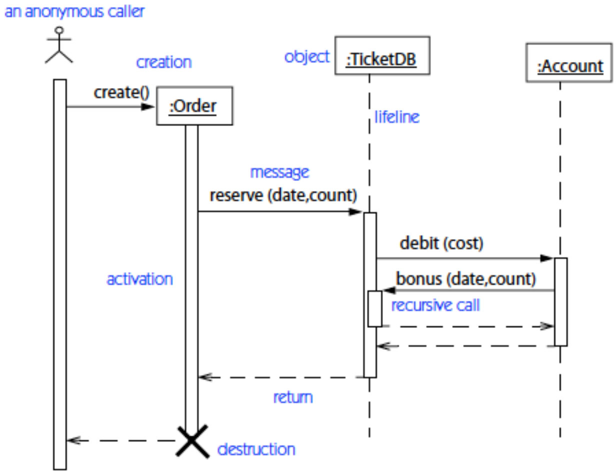
Sequence diagram

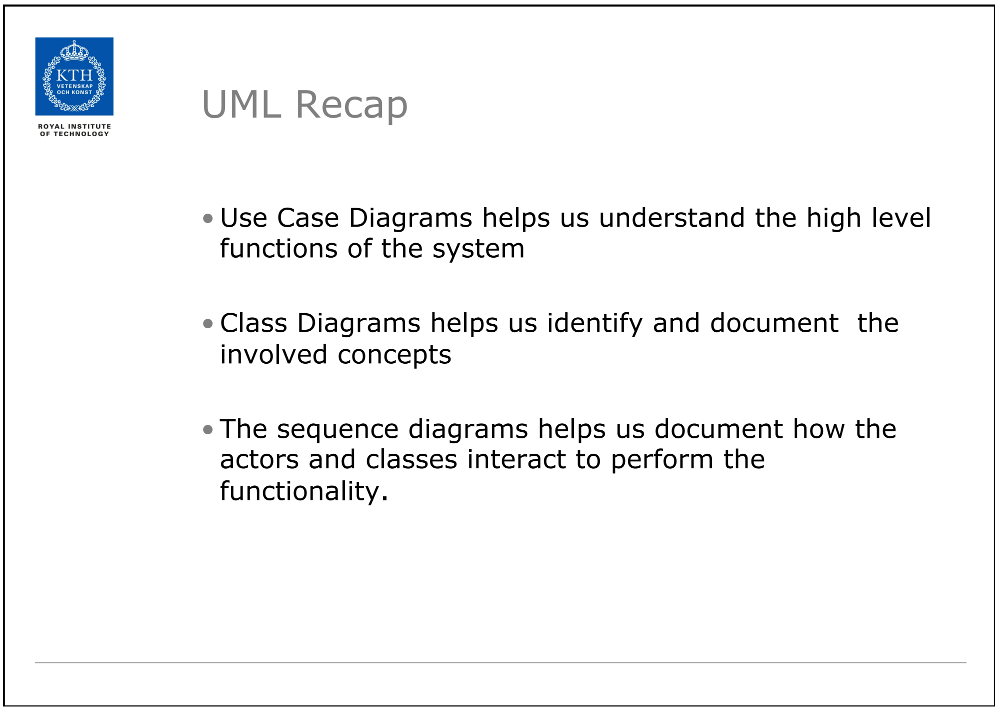
- Actors or Classes(objects)



Source: Unified Modeling Language Reference Manual, ISBN 0-201-30998-X, Addison-Wesley

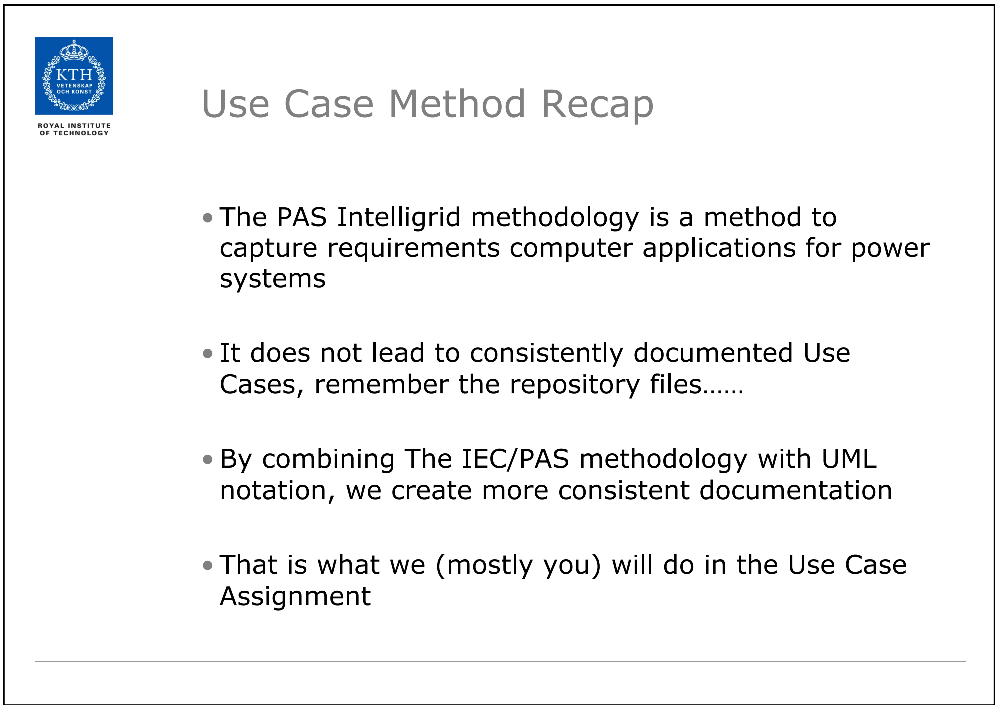
Activation & Destruction





UML Recap

- Use Case Diagrams helps us understand the high level functions of the system
- Class Diagrams helps us identify and document the involved concepts
- The sequence diagrams helps us document how the actors and classes interact to perform the functionality.



Use Case Method Recap

- The PAS Intelligrid methodology is a method to capture requirements computer applications for power systems
- It does not lead to consistently documented Use Cases, remember the repository files……
- By combining The IEC/PAS methodology with UML notation, we create more consistent documentation
- That is what we (mostly you) will do in the Use Case Assignment