EH2750 Computer Applications in Power Systems, Advanced Course.

Lecture 2
31 August 2011

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

Outline of the lecture

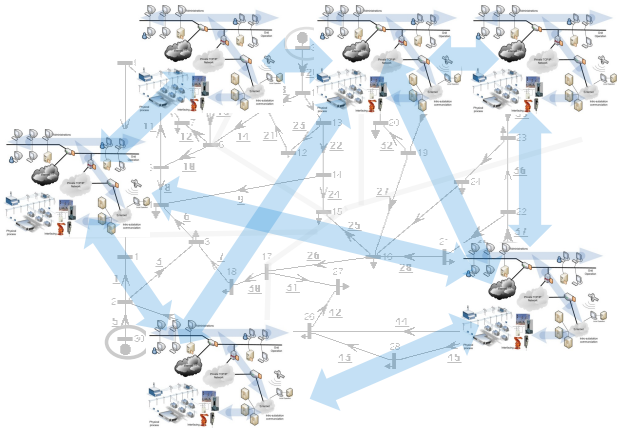
Smartgrids Use Case Methodology
The Smartgrid Use Case repository
UML – The Unified Modeling Language
What is UML
UML Diagrams
Break
UML Class Diagrams
UML Use Case Diagrams
UML Sequence Diagrams

How to know what to build?

"Who" controls what in a distribution system?
Voltage?
Frequency?
Is there a price for storage?
Can production be curtailed?
Can the system supply itself?
Can the DSO shift load in time?

Is the ICT architecture secure? *em architecture*
Is the performance sufficient?
Are the measurements of high quality? *ntrol*
Can all the systems communicate?
ɜ?

Integrated^2 Infrastructures



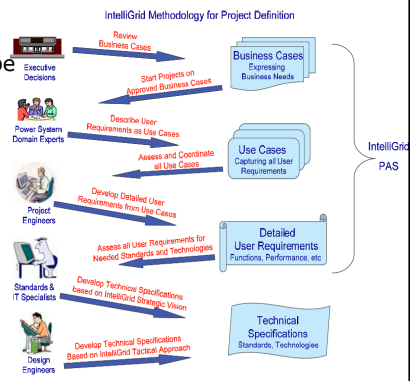
- Overarching issues:
  - Latency
  - Interoperability
  - Cybersecurity
  - Reliability

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.

The Intelligrid method

- **Phase 1:** Executives use Business Cases to approve projects in order to meet Business
- **Phase 2:** Domain Expert Stakeholders describe their User Requirements through the formal Use Case process
- **Phase 3:** Project Engineers develop the more detailed functional and performance requirements from the Use Cases
- **Phase 4:** Project Engineers and IT Specialists assess applicability to the project of the standards, technologies, and best practices
- **Phase 5:** Design Engineers develop Technical Specifications based on Strategic Vision, Tactical Approach, & Standards



Source: IntelliGrid Methodology for Developing Requirements for Energy Systems IEC/PAS 62559

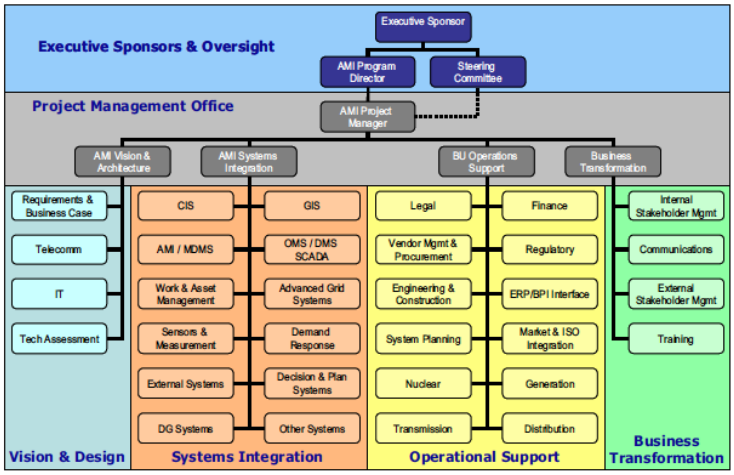
Phase 2 – A detailed look

Definition of a Use Case: Class specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system IEC 62390, ed. 1.0 (2005-01)

- **Step 1:** Identification of All Potential Stakeholders
- **Step 2:** Reviewing existing Architecture Use Cases
- **Step 3:** Brainstorming List of Functions (Use Cases) with Stakeholders
- **Step 4:** Drafting Use Cases
- **Step 5:** Reviewing and Updating Use Cases

Stakeholder identification

- Example from AMI project
- Stakeholders are critical for requirements capture and project acceptance

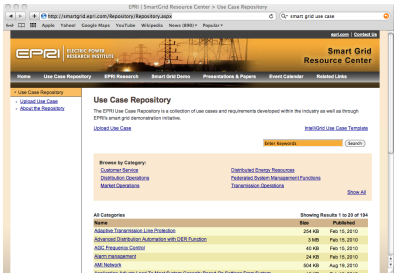


Source: IntelliGrid Methodology for Developing Requirements for Energy Systems IEC/PAS 62559

ROYAL INSTITUTE OF TECHNOLOGY

Step 2 – Review existing Use Cases

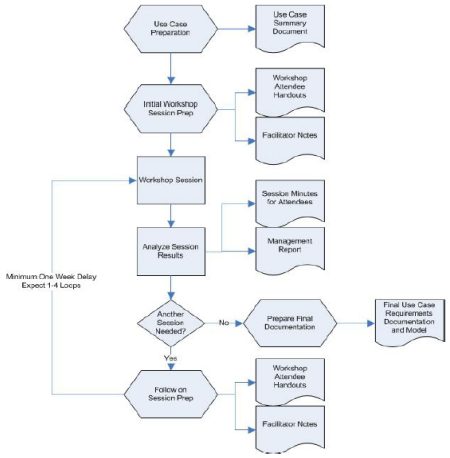
- The EPRI Use Case Repository
  - A huge collection of Use Cases exist online
  - Developed initially in the Intelligrid project in the late 1990s and continued during 2008+
  - The level of granularity & completeness very varied



ROYAL INSTITUTE OF TECHNOLOGY

Step 3 - Brainstorming

- Workshop format
- All stakeholders invovled
- Open discussions
- Documentation and analysis after the workshop

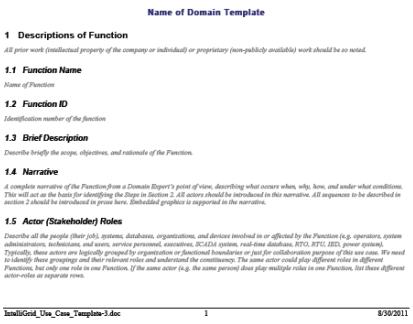


Step 4 – Drafting Use Cases

- 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.

Step4 -continued

- First draft of Use Case is documented in the Intelligrid Use Case Template



Name of Domain Template

1 Descriptions of Function

All prior work (intellectual property of the company or individual) or proprietary (non-publicly available) work should be so noted.

1.1 Function Name

Name of Function

1.2 Function ID

Identification number of the function

1.3 Brief Description

Describe briefly the scope, objectives, and rationale of the Function.

1.4 Narrative

A complete narrative of the Function from a Domain Expert's point of view, describing what occurs when, why, how, and under what conditions. This will act as the basis for identifying the Steps in Section 2. All actors should be introduced in this narrative. All sequences to be described in section 2 should be introduced in gross here. Embedded graphics is supported in the narrative.

1.5 Actor (Stakeholder) Roles

Describe all the people (their job), systems, databases, organizations, and devices involved in or affected by the Function (e.g. operators, system administrators, technicians, end users, service personnel, executives, SCADA system, real-time database, RTO, RTU, IED, power system). Typically, these actors are logically grouped by organization or functional boundaries or just for collaboration purpose of this use case. We need to identify these groupings and their relevant roles and understand the constituency. The same actor could play different roles in different Functions, but only one role in one Function. If the same actor (e.g. the same person) does play multiple roles in one Function, list these different actor-roles as separate rows.

Step 5 – Reviewing & updating

- Reviewing & updating continues as more information is gained.
- In Phases 3 & 4 the non-functional requirements are further detailed.

Items to consider for detailing

- Configuration
- Quality of Service Requirements
- Security Requirements
- Data Management Issues
- Constraints or Other Issues

The PAS provides checklists for all these areas

Outline of the lecture

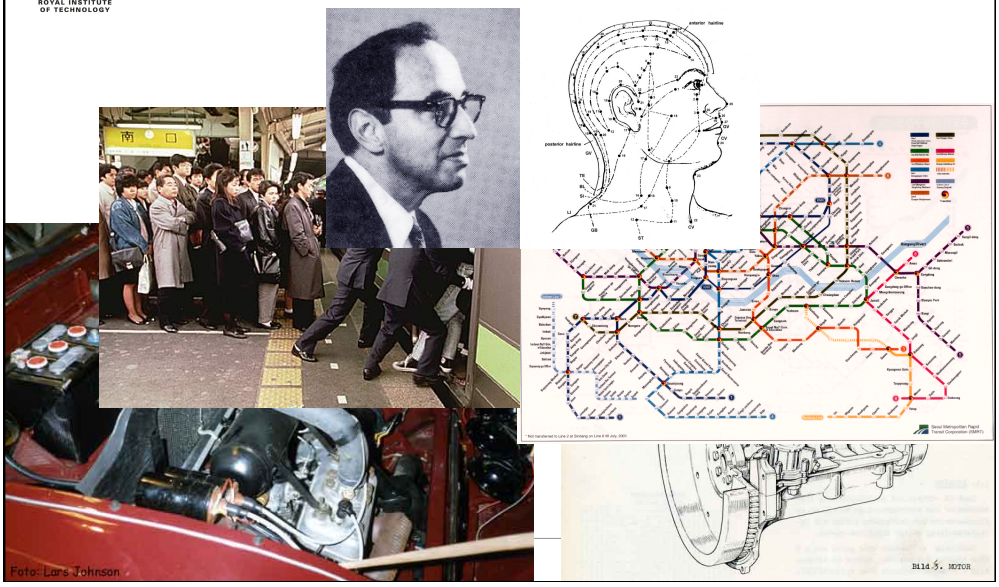
Smartgrids Use Case Methodology
- The Smartgrid Use Case repository

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

Break

UML Class Diagrams
UML Use Case Diagrams
UML Sequence Diagrams

How to make things understandable



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 .

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.

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

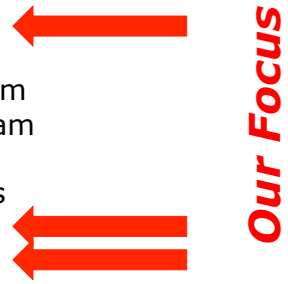
- Model management describes the organization of the models themselves into hierarchical units. The package is the generic organizational unit for models. Special packages include models and subsystems. The model management view crosses the other views and organizes them for development work and configuration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

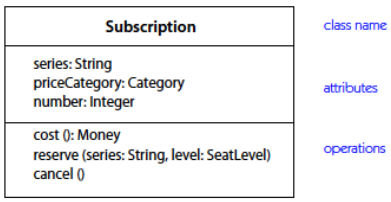
- Structural diagrams
  - Class diagram
  - Object diagram
  - Component diagram
  - Deployment diagram
  - Package diagram
- Behavioral diagrams
  - Use Case diagram
  - Sequence diagram
  - Activity diagram
  - Collaboration diagram
  - State chart



Our Focus

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*

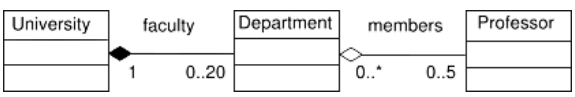
Relationship	*Function*	*Notation*
association	A description of a connection among instances of classes	———
dependency	A relationship between two model elements	- - - -→
flow	A relationship between two versions of an object at successive times	- - - -→
generalization	A relationship between a more general 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

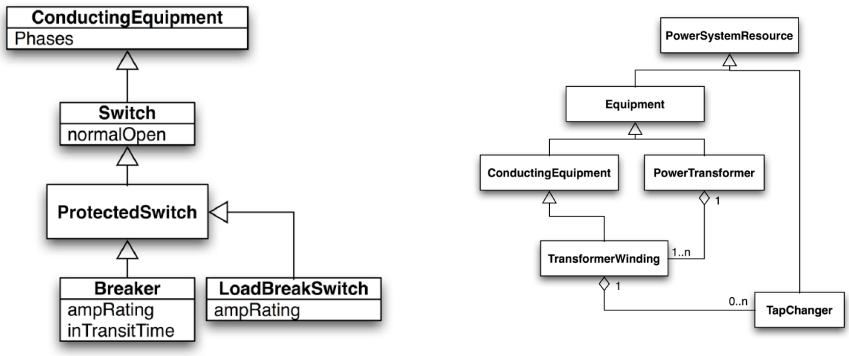
ROYAL INSTITUTE OF TECHNOLOGY

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



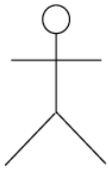
Source: An Introduction to IEC 61970-301 & 61968-11:The Common Information Model, A. McMorran, University of Strathclyde

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

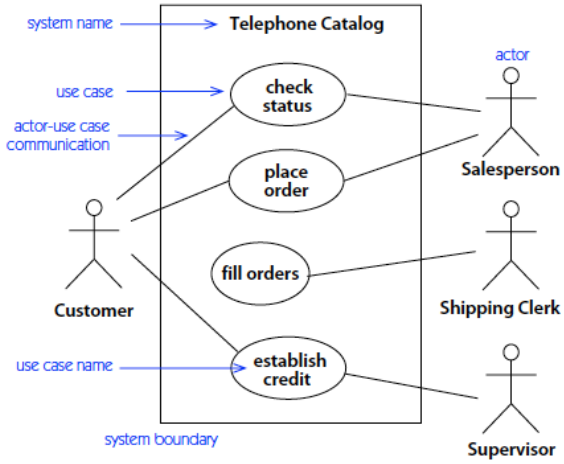
Actors in around the system



- 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



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

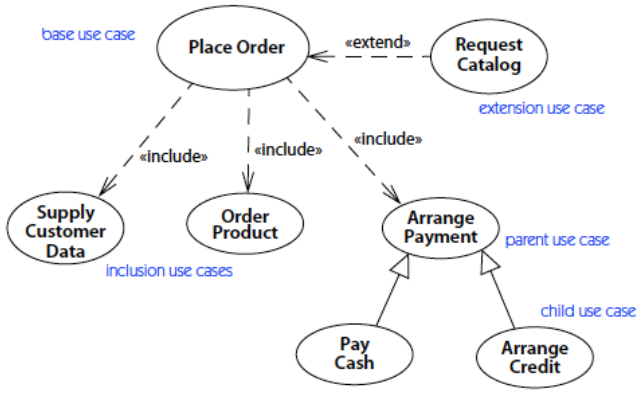
Use Case relations

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.



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

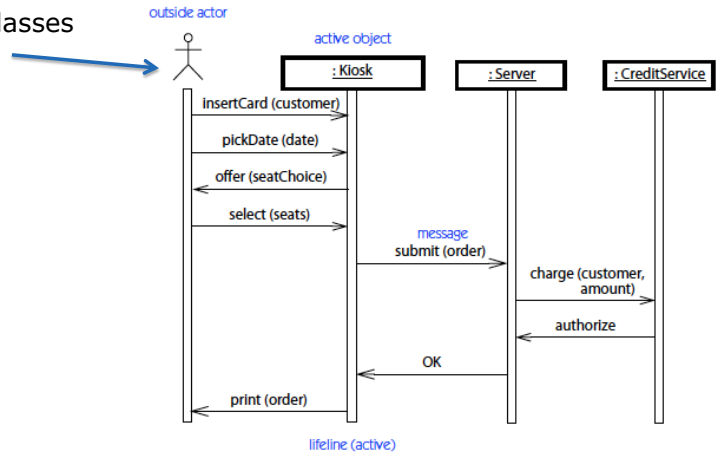
Sequence Diagrams

- 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

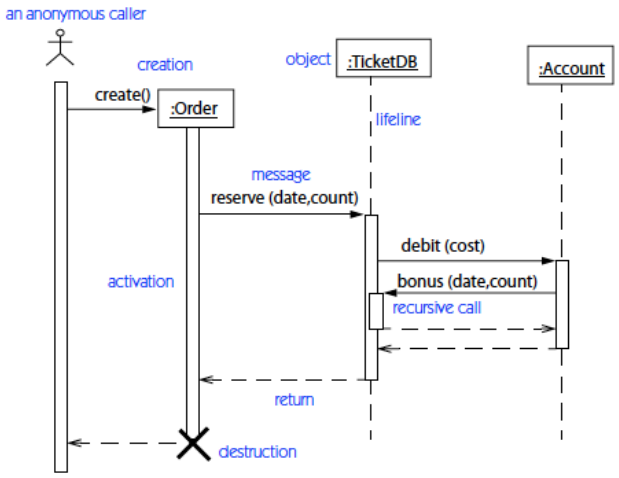
Sequence diagram

- 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