# Lecture #14
# Course Wrap-up

EH2745 Computer Applications in Power Systems
Introductory Course

## Agenda

- iTesla
- Java programming
- Information modelling
- Relational Databases
- Machine Learning

## Course Philosophy

The course has two (conflicting) aims

1. Develop the student as a programmer
2. Develop the student in Machine learning and data analysis for power system decision making

Why conflicting?

## Course Philosophy

We think you may have taken programming courses before
We think you may know something about information modeling
We think you may know something about data analysis & statistics
**If you do not, we will teach you the basics**

Power System data modeling | Machine Learning

Software Development in Java

We want you to combine these skills in this applied course

## Agenda

- Java programming
- Information modelling
- Relational Databases
- Machine Learning

## Java programming – no point to repeat

MIT slides for Java syntax and Good Programming Style

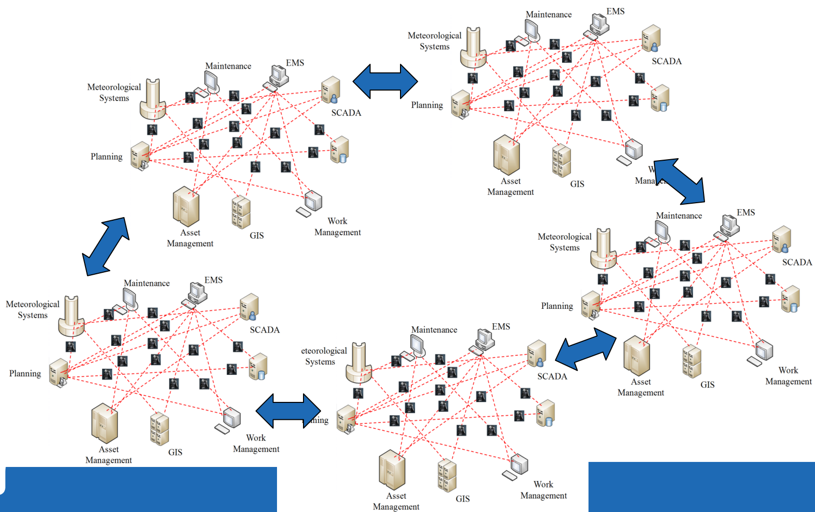
For the Test, be prepared to:

- Find errors in small Java programs
- Predict output from Java code examples
- Write/complete small pieces of Java code

# Agenda

- Java programming
- Information modelling
- Relational Databases
- Machine Learning

# Cross-utilities Information Exchange for planning and operation



## Which is the better  choice?

- **Comma Separated Values**
  - Pros
    - Little extra data has to be sent (only the commas)
  - Cons:
    - Data has to arrive in the right order
- **Using  Tags(XML)**
  - Pros:
    - Flexible format (data can arrive out of order)
    - Thanks to tags we can search for data
    - People can read it!!!
  - Cons:
    - Verbose – a lot of overhead!

## A Simple XML Document

Start Tag

```
<article>
  <author>Lars Nordström</author>
  <title>Meaning of Life</title>
  <text>
    <abstract>To begin with….<abstract>
    <section number="1" title="Introduction">
      The <index>meaning</index> of life is..
    </section>
  </text>
</article>
```

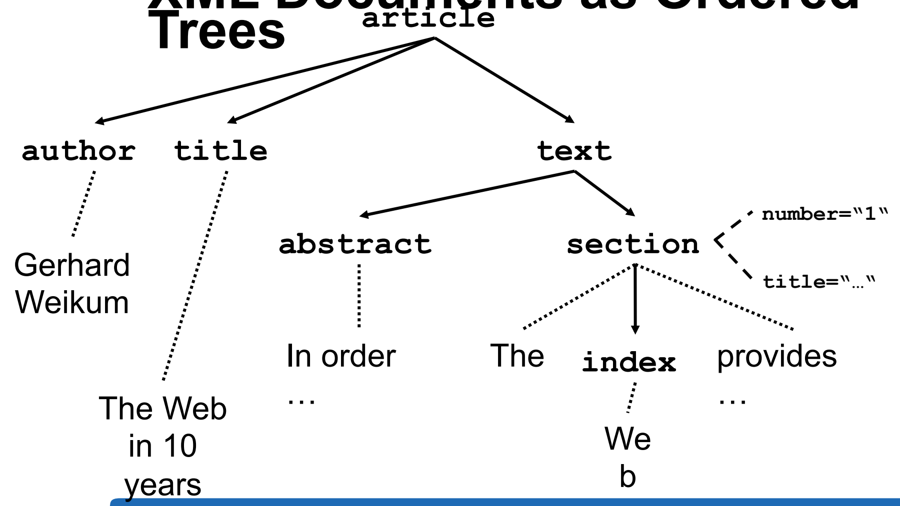
Attribute

Elements

End Tag

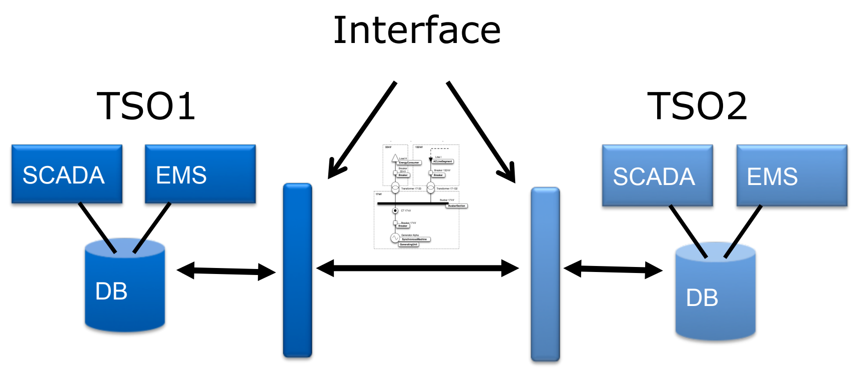
Source: XML for Beginners, Ralf Schenkel April 2003.

## XML Documents as Ordered Trees






## Data exchange architecture




Power system model transfer between TSOs

## RDF

RDF is ”language” to describe relation between things.

It is based on the format:

`Subject Predicate Object`

For example

Line is connected to busbar

## RDF continued

Consider the following Example:
Library data encoded in XML

```
<library name=”Glasgow Library”>
  <book title=”History of Glasgow, 1900-1950” author=”Walter Hannah”>
    <position section=”A” shelf=”2”/>
  </book>
  <book title=”A Brief History of Time” author=”Stephen Hawking”>
    <position section=”E” shelf=”4”/>
  </book>
  <book title=”History of Glasgow, 1950-2000” author=”Walter Hannah”>
    <position section=”A” shelf=”2”/>
  </book>
</library>
```

How to specify that Hannah’s books are related?

## RDF continued

By allowing relation between XML nodes (elements) relations can be described

A key requirements is of course that nodes (elements) are uniqly identifiable – this can be achieved by Name spaces and URIs

URIs are pointers to unique identifers of tags. In a way the URI is the uniquness, it may point to nothing.

## But where does the RDF Schema file come from?

**From an Information model!**

## What is the CIM?

A Unified Modeling Language (UML) based information model representing real-world objects and information entities exchanged within the value chain of the electric power industry

A tool to enable integration and information exchange to enable data access in a standard way

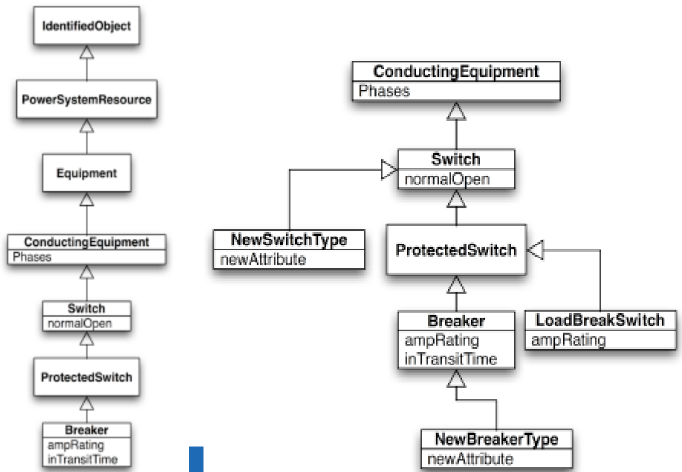
A common language to navigate and access complex data structures in any database

It is not tied to a particular vendor's view of the world

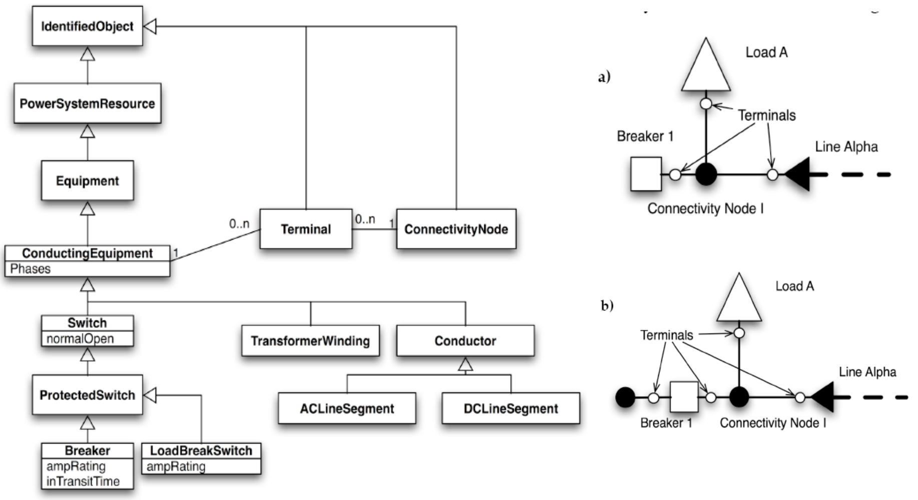
It also provides consistent view of the world by operators regardless of which application user interface they are using

*Courtesy of T.Saxton TC57 WG13 chairman*

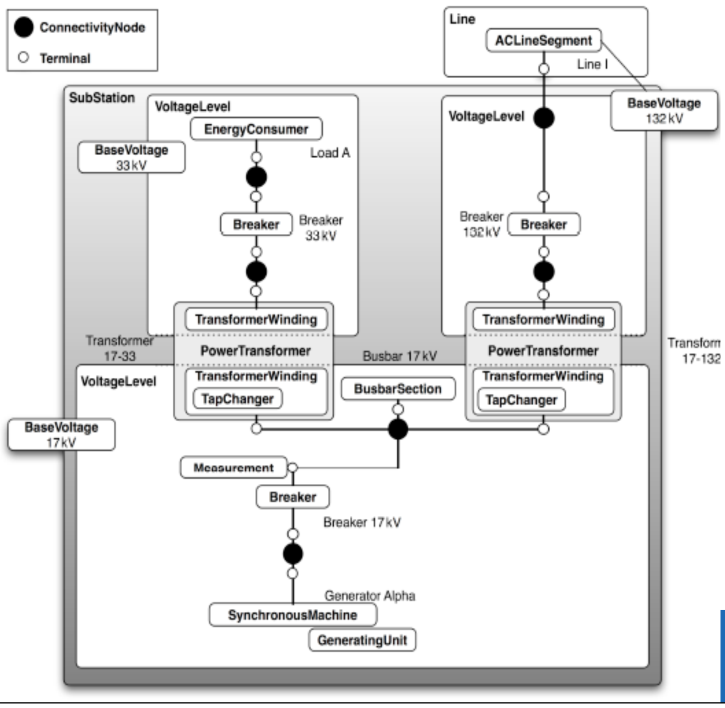
## Example Hierarchy in the Breaker Class



## Conducting Equipment and Connectivity



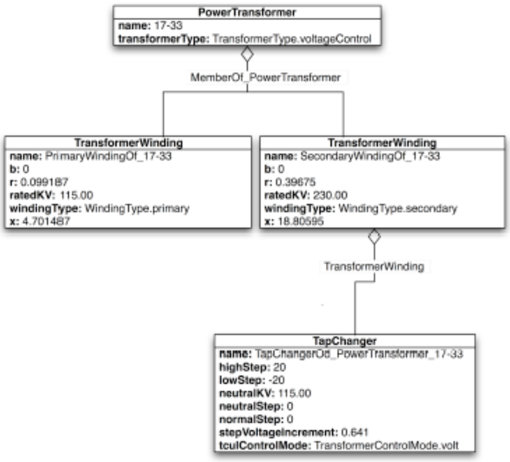

Conducting Equipment and Connectivity class diagram



## Now we can “define” the CIM RDFSchema

```
<rdfs:Class rdf:ID="PowerSystemResource">
  <rdfs:label xml:lang="en">PowerSystemResource</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Naming"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Equipment">
  <rdfs:label xml:lang="en">Equipment</rdfs:label>
  <rdfs:subClassOf rdf:resource="#PowerSystemResource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="ConductingEquipment">
  <rdfs:label xml:lang="en">ConductingEquipment</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Equipment"/>
</rdfs:Class>
```

## CIM RDF example



# CIM RDF example continued

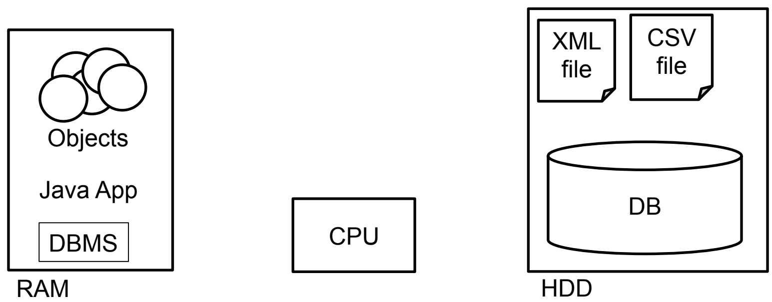
```
<rdf:RDF xmlns:cim="http://iec.ch/TC57/2003/CIM-schema-cim10#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

<cim:PowerTransformer rdf:ID="PowerTransformer_1733">
  <cim:PowerTransformer.transformerType
rdf:resource="http://iec.ch/TC57/2003/CIM-schema-
cim10#TransformerType.voltageControl"/>
  <cim:Naming.name>17-33</cim:Naming.name>
</cim:PowerTransformer>

<cim:TransformerWinding rdf:ID="PrimaryWindingOf_PowerTransformer_1733">
  <cim:TransformerWinding.b>0</cim:TransformerWinding.b>
  <cim:TransformerWinding.r>0.099187</cim:TransformerWinding.r>
  <cim:TransformerWinding.ratedKV>115.00</cim:TransformerWinding.ratedKV>
  <cim:TransformerWinding.windingType
rdf:resource="http://iec.ch/TC57/2003/CIM-schema-cim10#WindingType.primary"/>
  <cim:TransformerWinding.x>4.701487</cim:TransformerWinding.x>
  <cim:TransformerWinding.MemberOf_PowerTransformer
rdf:resource="#PowerTransformer_302"/>
  <cim:Naming.name>PrimaryWindingOf_17-33</cim:Naming.name>
</cim:TransformerWinding>
```

```
<cim:TransformerWinding rdf:ID="SecondaryWindingOf_PowerTransformer_1733">
  <cim:TransformerWinding.b>0</cim:TransformerWinding.b>
  <cim:TransformerWinding.r>0.39675</cim:TransformerWinding.r>
  <cim:TransformerWinding.ratedKV>230.00</cim:TransformerWinding.ratedKV>
  <cim:TransformerWinding.windingType
rdf:resource="http://iec.ch/TC57/2003/CIM-schema-
cim10#WindingType.secondary"/>
  <cim:TransformerWinding.x>18.80595</cim:TransformerWinding.x>
  <cim:TransformerWinding.MemberOf_PowerTransformer
rdf:resource="#PowerTransformer_302"/>
  <cim:Naming.name>SecondaryWindingOf_17-33</cim:Naming.name>
</cim:TransformerWinding>

<cim:TapChanger rdf:ID="TapChangerOf_PowerTransformer_1733">
  <cim:TapChanger.highStep>20</cim:TapChanger.highStep>
  <cim:TapChanger.lowStep>-20</cim:TapChanger.lowStep>
  <cim:TapChanger.neutralKV>115.00</cim:TapChanger.neutralKV>
  <cim:TapChanger.neutralStep>0</cim:TapChanger.neutralStep>
  <cim:TapChanger.normalStep>0</cim:TapChanger.normalStep>
  <cim:TapChanger.stepVoltageIncrement>0.641</cim:TapChanger.stepVoltageIncre
ment>
  <cim:TapChanger.tculControlMode rdf:resource="http://iec.ch/TC57/2003/CIM-
schema-cim10#TransformerControlMode.volt"/>
  <cim:TapChanger.TransformerWinding
rdf:resource="#PrimaryWindingOf_PowerTransformer_302"/>
  <cim:Naming.name>TapChangerOf_PowerTransformer_17-33</cim:Naming.name>
</cim:TapChanger>

</rdf:RDF>
```

## Agenda

- Java programming
- Information modelling
- Relational Databases
- Machine Learning

## Storing data persistently




During execution, RAM is used to store our data
Files can be read and write for persistent storage
But what if we want to access the data in a more flexible way?
Reading single posts, adding data, removing data etc.

## Relational data storage

Data is organised in tables of two dimensions
Rows & Columns

Tables are known as ”Relations”
Rows are ”Tuples”
Columns are ”Attributes”

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 3 | Steve | C |

## Cardinality & Degree

Attributes

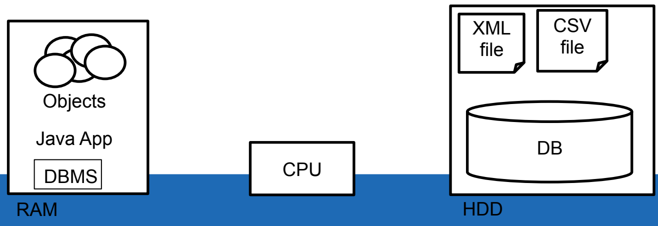
Tuples

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 3 | Steve | C |

The cardinality of a Relation is its number of tuples (rows)
The degree of a Relation is its number of attributes (columns)

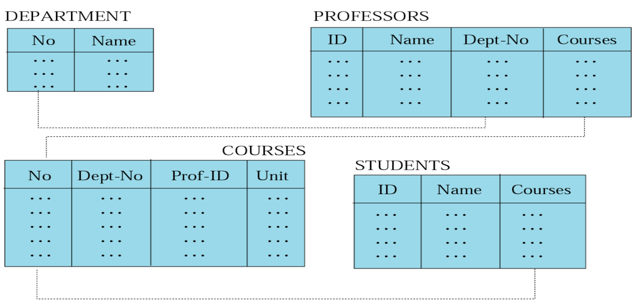
## What are these tables really?

1. Data is stored on the computer's HDD as bits (of course)
2. The data is struuctred according to some scheme that is efficient for the disk and CPU's access to the data
3. When we people want to write a (Java) program to manipulate the data, we think of it, and access it in the form of tables
4. The DBMS program translate from the tables to actual data storage (which is logical to the CPU but not to us)




## Entity Relationship Diagrams

"Relations between Relations"




By defining attributes as "Keys" we can relate Tuples from different Relations to each other.

## Good relations

Is this a good relation?

| Part | Qt | Warehouse | Adress |
|---|---|---|---|
| Wheel | 23 | Building2 | Main St 12 |
| Wheel | 12 | Building1 | Diagon Alley 3 |
| Seat | 9 | Building1 | Diagon Alley 3 |

Is this a good relation?

## E-R Diagrams ”must” be Normalised

- Normalisation of E-R diagrams is like ”Good Programming Style” but for Data
- It enables more efficient access to data and more efficient storage
- Reduces the risk of error in data.

- In Theory 5 levels of Normality (or Normal forms) exist
  - 1st Normal form
  - 2nd Normal form
  - 3rd Normal Form ← We stop here
  - 4th Normal Form
  - 5th Normal Form

## First Normal Form

The First Normal is basic housekeeping.

- All Tuples in a Relation must have the same number of attributes.
- Or The degree of all Tuples must be the same.

This borders on the obvious under the definition of a Relational Database, since this is the definition of a Relation

## Second Normal Form

Only relevant when the keys are composite, i.e., consists of several attributes

To fulfill Second normal form non-key fields cannot have facts about a part of a key.

| Warehouse | Part | Adress | Qt |
|---|---|---|---|
| Building2 | Wheel | Main St 12 | 23 |
| Building1 | Wheel | Diagon Alley 3 | 12 |
| Building1 | Seat | Diagon Alley 3 | 9 |

Keys

## Normalised to 2nd Normal Form

| Warehouse | Part | Qt |
|---|---|---|
| Building2 | Wheel | 23 |
| Building1 | Wheel | 12 |
| Building1 | Seat | 9 |

Keys

| Warehouse | Adress |
|---|---|
| Building1 | Diagon Alley 3 |
| Building2 | Main St 12 |

## Third Normal Form

In Third Normal Form, a non-key attribute must not hold information about another non-key attribute

| Course | Professor | Office |
|---|---|---|
| EH2745 | Nordström | Osquldas väg 10, floor 7 |
| EH2751 | Nordström | Osquldas väg 10, floor 7 |
| EJ2301 | Soulard | Teknikringen 33, floor 1 |
| EG2200 | Amelin | Reknikringen 35, floor 2 |

Key

## Normalised to 3rd Normal form

| Course | Professor |
|---|---|
| EH2745 | Nordström |
| EH2751 | Nordström |
| EJ2301 | Soulard |
| EG2200 | Amelin |

| Professor | Office |
|---|---|
| Nordström | Osquldas väg 10, floor 7 |
| Amelin | Teknikringen 33, floor 2 |
| Soulard | Teknikringen 33, floor 1 |

## Tuple Relational Calculus

With the definitions (Relation, Tuple, Attribute) above we can define a number of basic operations on relations

```
Insert
Delete
Update
Select
Project
Join
Union
Intersection
Difference
```

# Insert

Insert is a unary operation – it operates on a single Relation
It adds a Tuple to a Relation

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 3 | Steve | C |

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 3 | Steve | C |
| 4 | Lars | A |

```
Insert t in R
```

# Delete

Delete is a unary operation – it operates on a single Relation
It deletes a Tuple fulfilling criteria from a Relation

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 3 | Steve | C |
| 4 | Lars | A |

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 4 | Lars | A |

```
Delete t where a=x from R
```

# Update

Update is a unary operation – it operates on a single Relation
It modifies an attribute in Tuple fulfilling criteria in a Relation

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 4 | Lars | A |

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 4 | Lars | E |

```
Update t.a2=data where t.a1=x in R
```

# Select

Select is a unary operation – it operates on a single Relation
The Select operation creates a new relation R2  from relation R1
The Tuples inR1 is a subset of R2

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 4 | Lars | E |

| ID | Name | Grade |
|---|---|---|
| 2 | Bob | B |
| 4 | Lars | E |

```
Select * from R1 where ID >1
```

# Project

Project is a unary operation – it operates on a single Relation
The Project operation creates a new relation R2 from relation R1
The Attributes in R1 is a subset of R2

| ID | Name | Grade |
|---|---|---|
| 1 | Jill | D |
| 2 | Bob | B |
| 4 | Lars | E |

| Name |
|---|
| Jill |
| Bob |
| Lars |

Project Name from R1

# Join

Join is a binary operation – it operates two Relations
The Join operation creates a new relation R3 from relations R1 & R2
Based on common attributes (keys)

| Course | Professor |
|---|---|
| EH2745 | Nordström |
| EH2751 | Nordström |
| EJ2301 | Soulard |
| EG2200 | Amelin |

X

| Professor | Office |
|---|---|
| Nordström | Osquldas väg 10, floor 7 |
| Amelin | Teknikringen 33, floor 2 |
| Soulard | Teknikringen 33, floor 1 |

Not Normalised??

| Course | Professor | Office |
|---|---|---|
| EH2745 | Nordström | Osquldas väg 10, floor 7 |
| EH2751 | Nordström | Osquldas väg 10, floor 7 |
| EJ2301 | Soulard | Teknikringen 33, floor 1 |
| EG2200 | Amelin | Reknikringen 35, floor 2 |

Intermediate result for analysis

## Union

A binary operation – it operates on two Relations R1 and R2
Creates a new relation R3 in which each tuple is either in R1, in the R2, or in both R1 and R2.
The two relations must have the same attributes.

| Warehouse | Adress |
|---|---|
| Building3 | King St 23 |
| Building4 | Queens Rd 2 |

+

| Warehouse | Adress |
|---|---|
| Building1 | Diagon Alley 3 |
| Building2 | Main St 12 |

| Warehouse | Adress |
|---|---|
| Building1 | Diagon Alley 3 |
| Building2 | Main St 12 |
| Building3 | King St 23 |
| Building4 | Queens Rd 2 |

## Intersection

A binary operation – it operates on two Relations R1 and R2
Creates a new relation R3 in which each tuple is in both R1 and R2.
The two relations must have the same attributes.

| Warehouse | Adress |
|---|---|
| Building1 | Diagon Alley 3 |
| Building4 | Queens Rd 2 |

| Warehouse | Adress |
|---|---|
| Building1 | Diagon Alley 3 |
| Building2 | Main St 12 |

| Warehouse | Adress |
|---|---|
| Building1 | Diagon Alley 3 |

## Agenda

- Java programming
- Information modelling
- Relational Databases
- Machine Learning

## Power Systems Analysis - traditionally

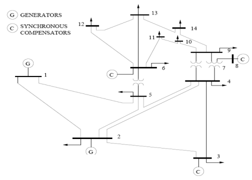
Power system analysis, control and operation is dependent on models

Using the models, analytical and numerical analysis provides decision support for e.g.

- Security
- Stability
- Optimal power flow
- Contingency analysis
- Expansion planning
- Market clearing

$$0 = -P_i + \sum_{k=1}^{N} |V_i||V_k|(G_{ik}\cos\theta_{ik} + B_{ik}\sin\theta_{ik})$$

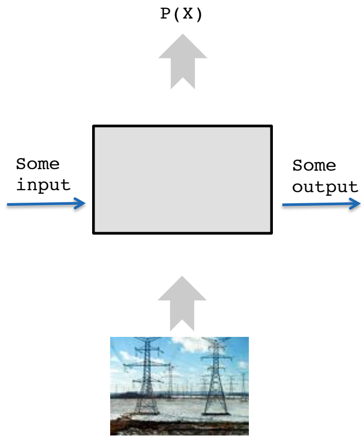
$$0 = -Q_i + \sum_{k=1}^{N} |V_i||V_k|(G_{ik}\sin\theta_{ik} - B_{ik}\cos\theta_{ik})$$




## Power Systems Analysis – An automated learning approach

Understanding states in the power system is established through observation of inputs and outputs without regard to the physical electrotechnical relations between the states.

Adding knowledge about the electrotechnical rules means adding heuristics to the learning.




*Given a set of examples (the learning set (LS)) of associated input/output pairs, derive a general rule representing the underlying input/output relationship, which may be used to explain the observed pairs and/or predict output values for any new unseen input.*

## Classes of methods for learning

In **Supervised** learning a set of input data and output data is provided, and with the help of these two datasets the model of the system is created.

For this introductory course, our focus is here With a short look at unsupervised learning

In **Unsupervised** learning, no ideal model is anticipated, but instead the analysis of the states is done in order to identify possible correlations bewteen datapoints.

In **Reinforced** learning, the model in the system can be gradually refined through means of a utility function, that tells the system that a certain ouput is more suitable than another.

## Classification vs Regression

Two forms of Supervised learning

**Classification:** The input data is number of switch operations a circuitbreaker has performed and tthe output is a notification whether the switch needs maintenance or not. ”Boolean”

**Regression:** Given the wind speed in a incoming weather front, the output is the anticipated production in a set of wind turbines. ”Floating point”

## Supervised learning - a preview

In the scope of this course, we will be studying three forms of supervised learning.

- Decision Trees
  *Overview and practical work on exercise session.*
- Artificial Neural Networks
  *Overview only, no practical work.*
- Statistical methods – k-Nearest Neighbour
  *Overview and practical work on exercise session. Also included in Project Assignment*
  *kNN algorithm can also be used for unsupervised clustering.*

## How to measure information content

Entropy $\mathbf{H}$ is a measure of *Unpredictability*.

Defined as:

$$-\sum p_i \log p_i$$

Where

$\mathrm{p_i}$ is the probability of event $\mathrm{i}$

## An example of classification entropy

| Color | Size | Shape | Eadible? |
|---|---|---|---|
| Yellow | Small | Round | Yes |
| Yellow | Small | Round | No |
| Green | Small | Irregular | Yes |
| Green | Large | Irregular | No |
| Yellow | Large | Round | Yes |
| Yellow | Small | Round | Yes |
| Yellow | Small | Round | Yes |
| Yellow | Small | Round | Yes |
| Green | Small | Round | No |
| Yellow | Large | Round | No |
| Yellow | Large | Round | Yes |
| Yellow | Large | Round | No |
| Yellow | Large | Round | No |
| Yellow | Large | Round | No |
| Yellow | Small | Irregular | Yes |
| Yellow | Large | Irregular | Yes |

*Source: F. Aiolli - Sistemi Informativi University of Padova*

## Entropy example

Entropy for the example data set is calculated as:

$$I(all\_data) = -\left[\left(\frac{9}{16}\right)\log_2\left(\frac{9}{16}\right) + \left(\frac{7}{16}\right)\log_2\left(\frac{7}{16}\right)\right]$$

Giving: 0,9836

Is this reasonable?

*Source: F. Aiolli - Sistemi Informativi University of Padova*

## Information Gain

The reduction in Entropy achieved by partitioning the dataset differently.
Lets separate for instance per the attribute Size.

Size

Small

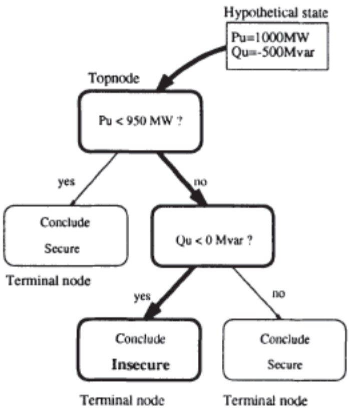
| Color | Size | Shape | Edible? |
|---|---|---|---|
| Yellow | Small | Round | + |
| Yellow | Small | Round | - |
| Green | Small | Irregular | + |
| Yellow | Small | Round | + |
| Yellow | Small | Round | + |
| Yellow | Small | Round | + |
| Green | Small | Round | - |
| Yellow | Small | Irregular | + |

Large

| Color | Size | Shape | Edible? |
|---|---|---|---|
| Green | Large | Irregular | - |
| Yellow | Large | Round | + |
| Yellow | Large | Round | - |
| Yellow | Large | Round | + |
| Yellow | Large | Round | - |
| Yellow | Large | Round | - |
| Yellow | Large | Round | - |
| Yellow | Large | Irregular | + |

*Source: F. Aiolli - Sistemi Informativi University of Padova*

## Back to our Power System example




Perhaps we can partition our dataset according to some attribute?

Lets try $P_u$ < 950MW

Equivalent If-Then rules :

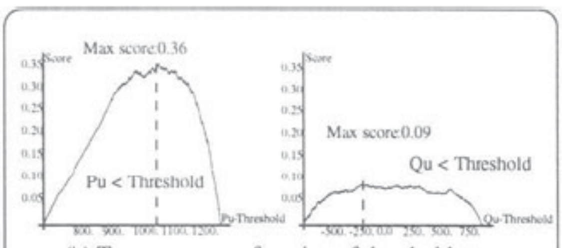
Rule 1 : If (Pu < 950MW) then Conclude Secure

Rule 2 : If (Pu > 950MW) and (Qu < 0Mvar) then Conclude Insecure

Rule 3 : If (Pu > 950MW) and (Qu > 0Mvar) then Conclude Secure

## Finding best partition.

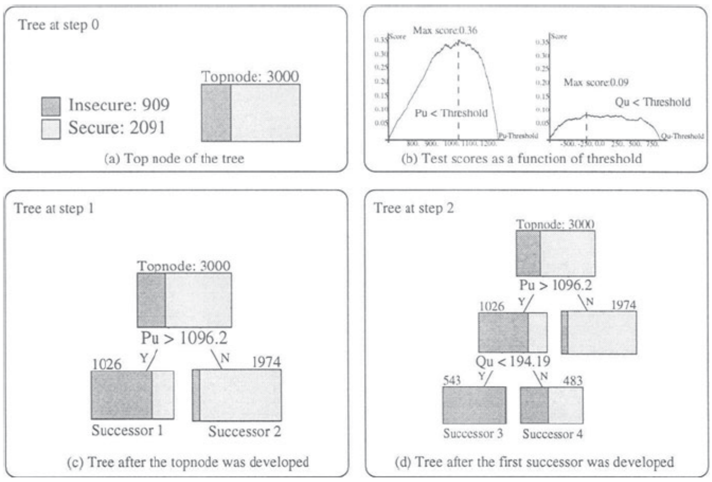
Starting with the candidate attributes (Pu and Qu) in our case
We check chich of the values for Pu and Qu that create the most valuable partition in terms of information gain.



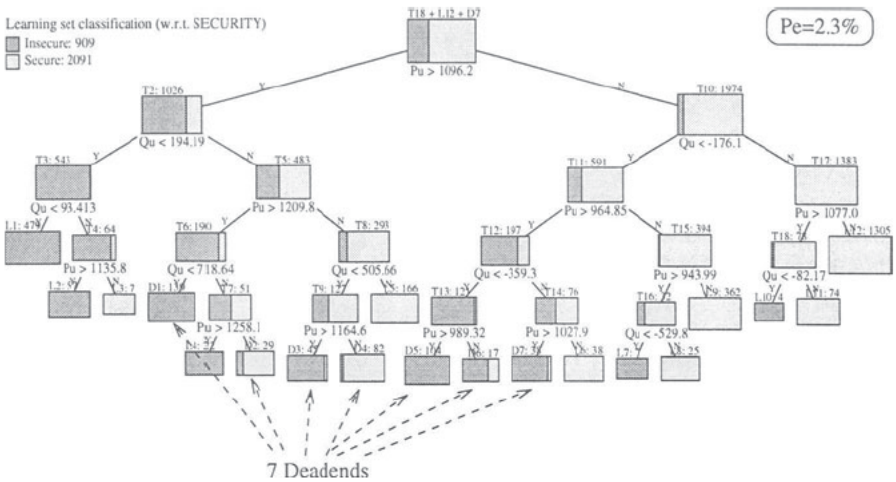

(b) Test scores as a function of threshold

Pu > 1096,2 MW is the best partition

## Gradual expansion of the Decision Tree




## Complete Decision Tree



## How to stop?

The splitting of data sets continues until either:

A perfect partition is reached – i.e. One which perfecly explains the content of the class – a *leaf*

One where no infomration is gained no matter how the data set is split. – a *deadend*.

## Validation of the Decision Tree

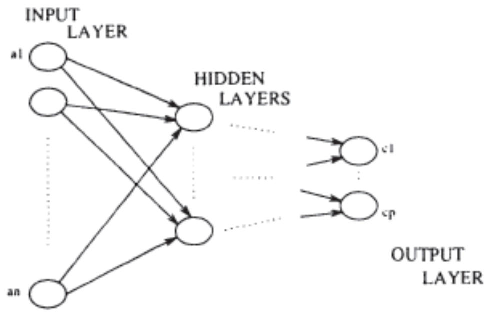
By using the Test Set (2000 samples) we can calidate the Decision tree.

By testing for each Object in the Test Set, we determine if the Decision tree provides the right answer for the Object.

In this particular example, the probability o error can be determined to 2,3. I.e. Of the 2000 samples 46 were classififed to the wrong class.

## Artificial Neural Network

Multi Layer Perceptrons (MLP)
A network of interconnected Perceptrons in several layers
First layer recives input, forwards to second layer etc.
Normally one hidden layer is sufficient to create good mappings




## Where is the "learning" in ANN

Given an input vector $a(o)$ (attributes of an object)

For a classification problem

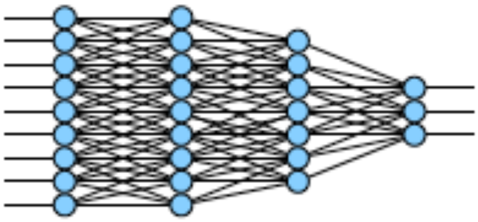
We want to assign it to a class $C_i$

For a regression problem

We want it to approximate a value $y$

We have to tune the weights of the inputs of the perceptrons

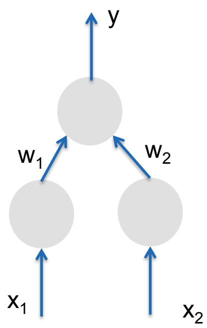
## So how to tune the weights in this …?



10s of perceptrons, 100s of links, 1000s of input values…

## Backpropagation algorithm

Trivial case



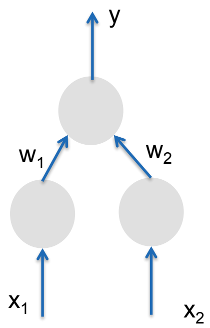

Remember, we are discussing **supervised** learning.

This means we have a sets of the following form: $(x_1,x_2,t)$

Input attributes and a correct target value t, that we want to achieve.

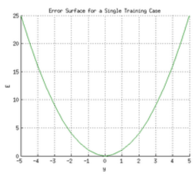
## Backpropagation algorithm

Trivial case




The Least Squares Error

$$E = (t - y)^2$$

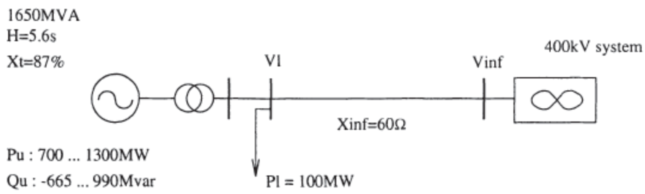


For a linear Perceptron

$$y = x_1 w_1 + x_2 w_2$$

Find minima of $E(y)$ w.r.t $(w_1, w_2)$

## Example from Automatic Learning techniques in Power Systems

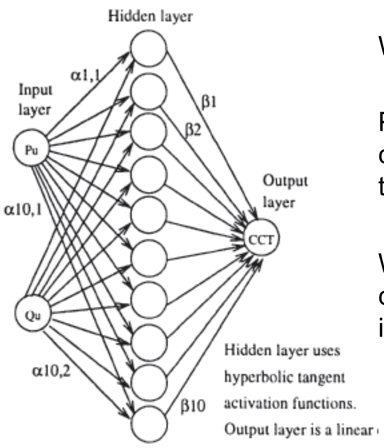



One Machine Infinite Bus (OMIB) system

- Assuming a fault close to the Generator will be cleared within 155 ms by protection relays
- We need to identify situations in which this clearing time is sufficient and when it is not
- Under certain loading situations, 155 ms may be too slow.

*Source: Automatic Learning techniques in Power Systems, L. Wehenkel*

## Initial ANN for the OMIB problem




Weights are random

Perceptrons use linear combination of inputs and tanh function

We want to calculate the clearing time (CCT), i.e. This is a **Regression** problem

$$\text{Output}_i(\text{state}) = \tanh(\alpha_{i,1} Pu(\text{state}) + \alpha_{i,2} Qu(\text{state}) + \theta_i),$$

## Output and Error function

The Output function is:

$$\text{CCT}_{\text{MLP}}(\text{state}) = \sum_{i=1\ldots 10} \beta_i \tanh(\alpha_{i,1} Pu(\text{state}) + \alpha_{i,2} Qu(\text{state}) + \theta_i),$$

The error function is:

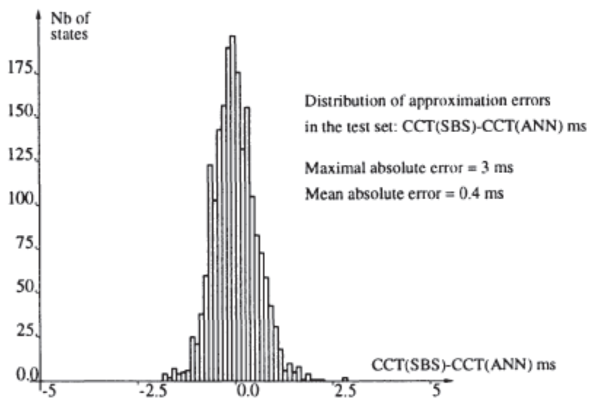
$$SE = N^{-1} \sum_{\text{state} \in \boldsymbol{LS}} |\text{CCT}(\text{state}) - \text{CCT}_{\text{MLP}}(\text{state})|^2,$$

## The final ANN structure is

After 46 iterations

$$
\begin{aligned}
\mathrm{CCT_{MLP}} = \; & -0.602710 \tanh(0.000194 Pu - 0.00034 Qu - 0.93219) \\
& -0.401320 \tanh(0.000822 Pu - 0.00020 Qu - 0.76681) \\
& +0.318249 \tanh(0.000239 Pu - 0.00050 Qu - 0.29351) \\
& -0.287230 \tanh(0.002004 Pu - 0.00034 Qu - 1.20080) \\
& +0.184522 \tanh(0.000131 Pu - 0.00057 Qu - 0.03152) \\
& +0.177701 \tanh(0.001799 Pu - 0.00011 Qu - 2.08190) \\
& -0.150720 \tanh(0.001530 Pu - 0.00056 Qu - 1.68040) \\
& +0.142678 \tanh(0.002152 Pu - 0.00046 Qu - 1.72280) \\
& -0.067897 \tanh(0.001910 Pu - 0.00051 Qu - 1.71343) \\
& -0.056020 \tanh(0.000202 Pu - 0.00085 Qu - 0.39876)
\end{aligned}
$$

## Error estimation with Test set

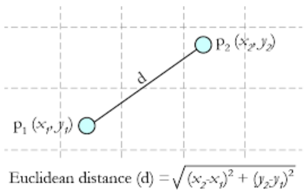


## The k Nearest Neighbour algorithm

The Nearest Neighbour algorithm is a way to classify objects with attributes to its nearest neighbour in the Learning set.

In k–Nearest Neighbour, the k nearest neighbours are considered.

”Nearest” is measured as distance in Euclidean space.



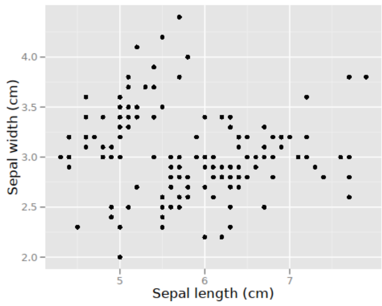

## K-means clustering

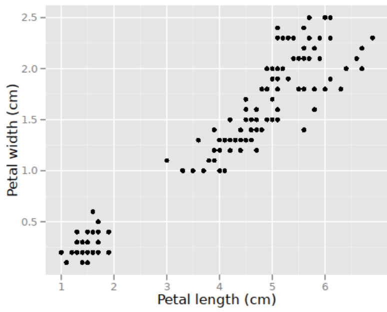
K-means clustering involves creating clusters of data
It is iterative and continues until no more clusters can be created
It requires the value of k to be defined at start.

Consider for instance a table like the following:

| Sepal length | Sepal width | Petal length | Petal width |
|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 |
| 4.9 | 3.0 | 1.4 | 0.2 |
| 4.7 | 3.2 | 1.3 | 0.2 |
| … | … | … | … |

## Plotted the data looks something like




## K-means clustering (continued)

In k means clustering, first pick k mean points randomly in the space

Caculate the distance from each object to the points

Assign datapoint to its closest mean point

Recalculate means

Once ended, we have k clusters

## k-Nearest Neighbour classification

Assuming instead a table like this where we have lables to ”clusters”

| Sepal length | Sepal width | Petal length | Petal width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | iris setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | iris setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | iris setosa |
| … | … | … | … | |
| 7.0 | 3.2 | 4.7 | 1.4 | iris versicolor |
| … | … | … | … | … |
| 6.3 | 3.3 | 6.0 | 2.5 | iris virginica |
| … | … | … | … | … |

## K-Nearest Neighbour algorithm

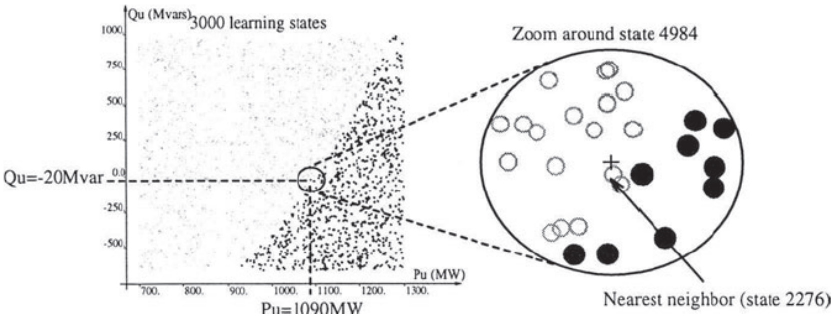
Given a new set of measurements, perform the following test:

Find (using Euclidean distance, for example), the k nearest entities from the training set. These entities have known labels. The choice of k is left to us.

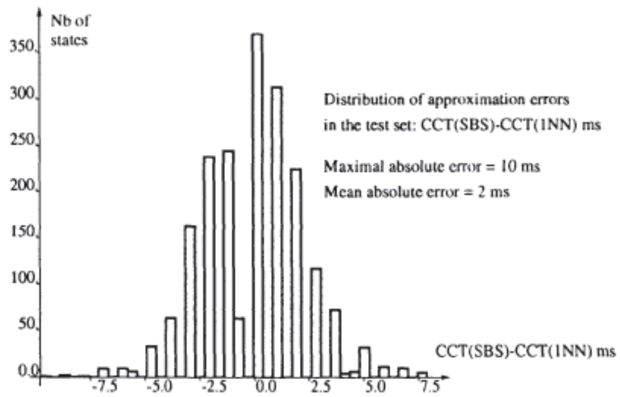
Among these k entities, which label is most common? That is the label for the unknown entity.

## In the OMIB example database

Sample 4984, and its neighbours




## Error in the 1-NN classification



## The Most important Slide – What's on the test

**Information Modeling:**
- Explain relation Information model <-> RDF
- Verify XML structure

**Relational Databases**
- Verify 1,2 & 3rd normal form
- Create E-R Diagrams
- Convert E-R diagrams into Tables & Attributes

**Machine Learning:**
- Create a Decision tree for a small dataset
- Explain ANN reasoning
- Explain kNN & k-means algorithms

**Simple ”computing by hand” questions may occur!**