Risto Kivioja

Section Manager, Embedded Software

Radio Access Systems

Contents:

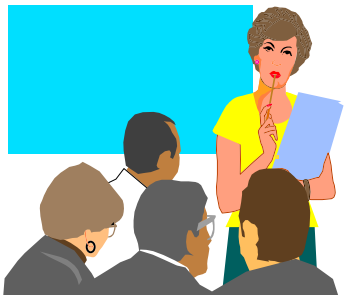
ERICSSON

Factors for successful SW development

ERICSSON

Factors for successful SW development
Skills and competence

- It's all about people



ERICSSON

Factors for successful SW development
Skills and competence

- Software design is about programming:

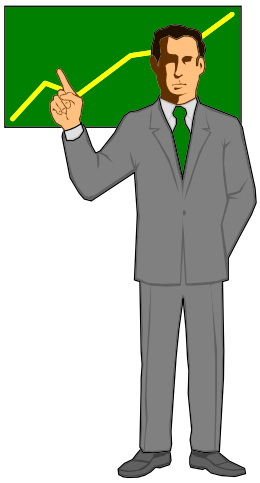
- Programming is:
  - Specification
  - Design
  - Coding
  - Testing
  - Experimenting
  - Maintenance

ERICSSON

Factors for successful SW development
Skills and competence

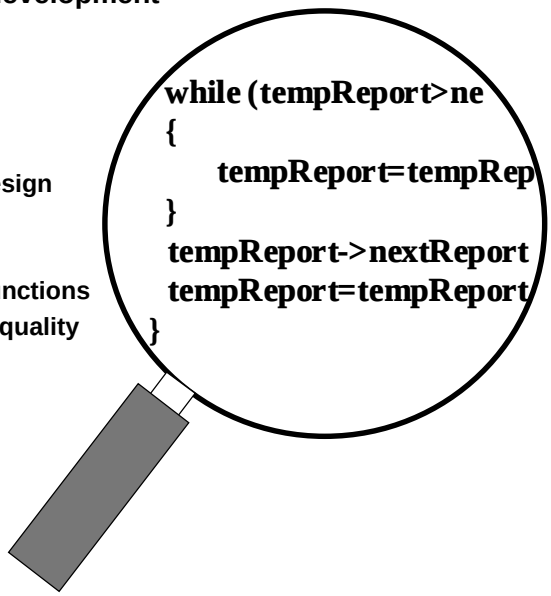
- Programmers are very skilled people
  - SW design is very hard work
  - Some people just can't program
  - Need proper education
- Lots of people avoid programming
- For some people it's easier to:
  - Be a manager
  - Make OH pictures
  - Go to meetings
  - Write reports
  - Hold lectures

  than to design and test software



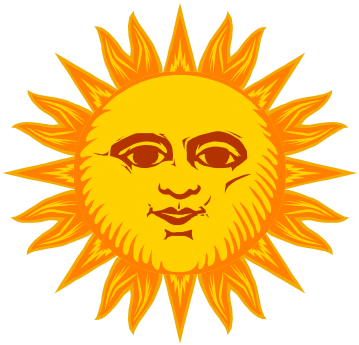
Factors for successful SW development
Skills and competence

- Need to focus on code
- Code is:
  - The output of software design
  - Is what you can sell
  - Is what you can measure
  - Is what implements the functions
  - Is what has an impact on quality



Factors for successful SW development
Skills and competence

- Happy programmers work best
- You have to give programmers:
  - Tools they want to use
  - Enough pay
  - Responsibility for their own work
  - Right to make decisions
  - Flexible time of work
  - Respect



ERICSSON

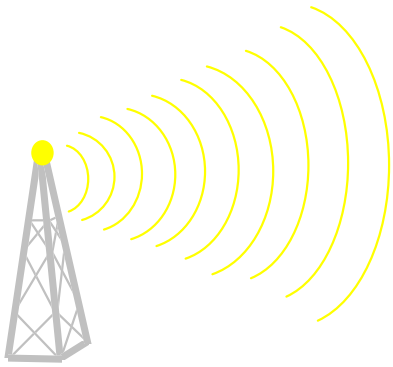
Factors for successful SW development
Skills and competence

- Learning programming is more than learning a language
- A skilled programmer knows about:
  - Operating systems
  - Languages
  - Algorithms
  - Specification methods
  - Testing tools and methods
  - Databases
  - IP
  - Distributed systems

ERICSSON

Factors for successful SW development
Skills and competence

- Knowledge of the application being developed is just as important as programming skills.



Factors for successful SW development
Skills and competence

- We must:
  - Employ programmers with the right education
  - Continuously educate our programmers
  - Use the best programmers for most difficult jobs
  - work in teams which is a good way to educate people

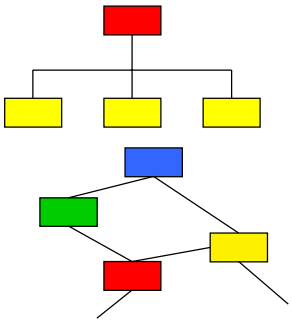
Factors for successful SW development
Skills and competence

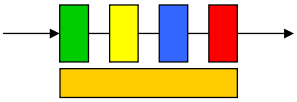
Programming is an Art

- The perfect programmer
  - Enjoys making software
  - Has the right sort of intelligence
  - Has broad computer science skills
  - Has all round ability (specs, design, testing, maintenance etc)
  - Has good knowledge of the application

Factors for successful SW development
SW Architecture

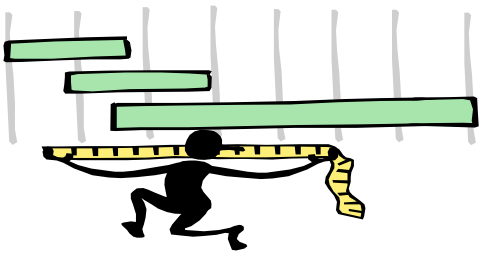
- You can't build a house without a plan, you can't build a new software system either
- SW architectures mean different things to different people:
  - Division of the system into parts (subsystems, blocks, modules etc)
  - Concurrency model
  - Distribution model
  - Failure / recovery model
  - Definition of interfaces





Factors for successful SW development
SW Architecture

- SW architecture can only be done by small number of people
- SW architecture designed by committees spell disaster



ERICSSON

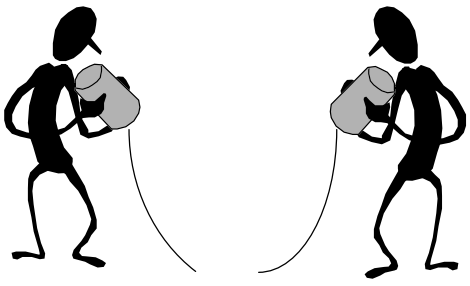
Factors for successful SW development
SW Architecture

- **Performance**
  - It is almost impossible to predict performance
  - Measurements on prototype systems is the only way
  - Concurrency and distribution model are the key factors
  - Performance can be tuned by prototyping

ERICSSON

Factors for successful SW development
SW Architecture

- **Interfaces**
  - Getting modularization and interfaces right requires experiments and prototypes
  - The prototyping environment and the final environment need not to be the same
  - Only small number of people should be involved



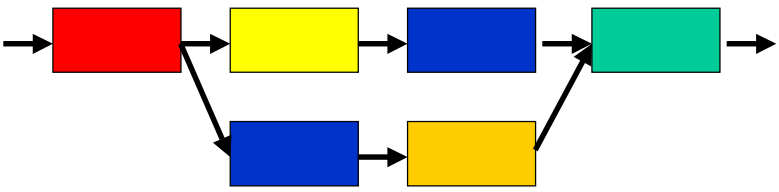
Factors for successful SW development
SW Architecture

SW Architecture is an Art

- The perfect architect
  - Has made several similar systems before
  - Has broad computer science skills and can make his own prototypes
  - Has deep knowledge of concurrency, distribution, methods
  - Has good knowledge of the application
  - Can build prototypes by himself
  - Is an artist

Factors for successful SW development
Organization and methods

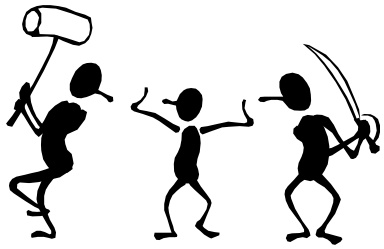
- Methods and processes are necessary
- Methods solve part of the problem not all of it



ERICSSON

Factors for successful SW development
Organization and methods

- New product development is different from maintenance and modification of existing products
- Developing products with substantial introduction of new software technology must take into account the learning process and the errors which will be made

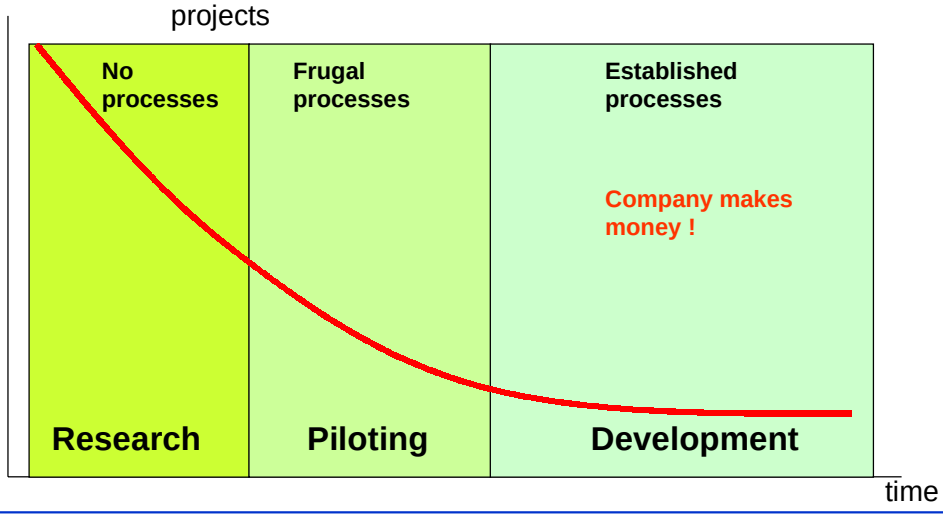


ERICSSON

Factors for successful SW development
Organization and methods R&P&D

cost/product

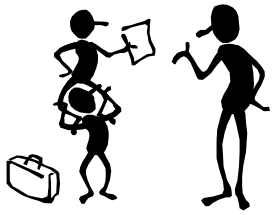
You can't have the same development process for all projects



ERICSSON

Factors for successful SW development
Organization and methods

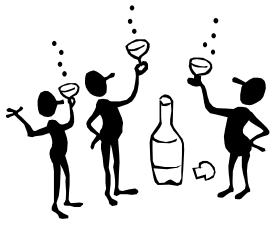
- Beware OH designers and paper studies when developing new products with new technology
  - Do it at one site under one roof
  - The developed process must be based on a series of prototypes and experiments
  - Until you are sure of what you are doing keep the number of people small
  - Use experienced programmers and architects



ERICSSON

Factors for successful SW development
Organization and methods

- Big Bang development processes are dead
- SW systems should be grown and evolved
- Start development with small team and grow slowly
- Adding manpower to a late SW projects makes it later
- Daily (or nightly) build combined with automatic regression testing
- When things go wrong consider all causes
  - Technology and tools
  - Architecture
  - Organization and methods
  - Skills and competence



ERICSSON

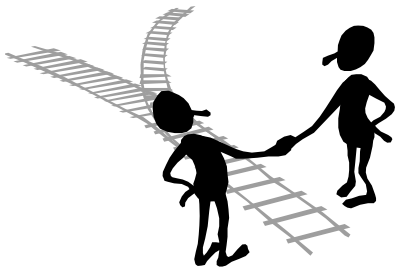
Factors for successful SW development
Organization and methods

- Different development process for different applications
  - A development process which is good for one technology or application, need not be the best for others
  - A development process which is good for one group of people, need not be the best for others

ERICSSON

Factors for successful SW development
Organization and methods

- One project - One goal
- Everyone involved to be working towards the same goal
- Configuration management, IS/IT support, administration, quality assurance etc are part of the development project and should be planned with the project



Factors for successful SW development
Organization and methods

SW Development methods are an Art

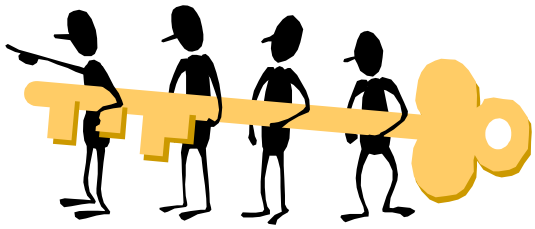
- A good development process:
  - Makes mistakes “in the small” - it allows for experiments and prototyping
  - Is an evolutionary process with daily builds and automatic regression testing
  - Is adapted to the technology, tools and people in the project
  - Gives honest and truthful reports about progress
  - Treats programmers and architects as artists and humans, not resources

Factors for successful SW development
Technology and Tools

- You need the best technology and tools for efficient software development

Factors for successful SW development
Technology and Tools

- Tools which are suitable for one application need not be suitable for another
- Tools which are suitable for one technology need not to be suitable for others
- The main thrust of technology and tools development is the desktop



Factors for successful SW development
Technology and Tools

- Introducing new tools and technology
  - Don't believe all the hype in the technical press
  - Base the choice of tools and technology on experiments and measurements
  - Introduce new technology and tools in small less critical projects and expand to larger ones later
  - The first program you write with a new tool / language / technology should be thrown away

Factors for successful SW development
Technology and Tools

- Don't stick too long with old technology
  - Plan for replacing or modernizing technology as new technology arrives
  - Don't let technology get so embedded in methods and processes that you can't change it
  - Make sure that new technology can work together with the old

Factors for successful SW development
Organization and methods

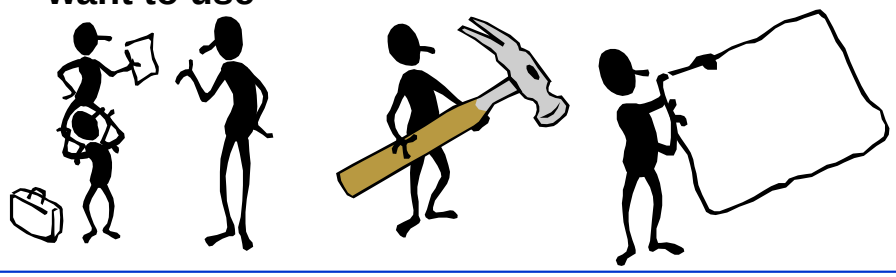
Tools and Technology is a Science

- Good tools and technology are:
  - Right for the application and people using them
  - Chosen as the result of measurements and experiments
  - Introduced in small "trial" projects before they are used in large ones
  - Modernized or replaced when better technology and tools become available
  - The tools and technology the programmers want to use

ERICSSON

Factors for successful SW development

- Committed and competent people
- Architecture done by small number of people
- Evolutionary development at one site
- Tool and technology what the programmers want to use



ERICSSON

SW Modeling

ERICSSON

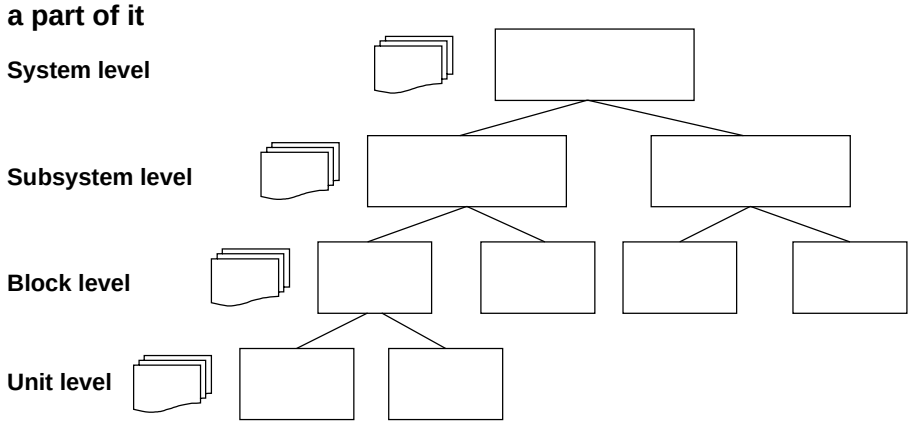
SW modeling
System Architecture

- You have to have a System Architecture
- The system architecture is a set of rules making up a conceptual framework. The rules control software and hardware design so that the system will have the desired system characteristics
- There are a number of reasons why it is good to have a System Architecture:
  - It is needed to conquer system complexity
  - It is needed to be able to have many people, possible of many locations, involved in a developing one large system
  - It is needed to give design stability to your system in the continuously changing world
  - It is needed a guidance in later detailed design decisions
  - It is needed to be able to maintain and evolve your system over long time

ERICSSON

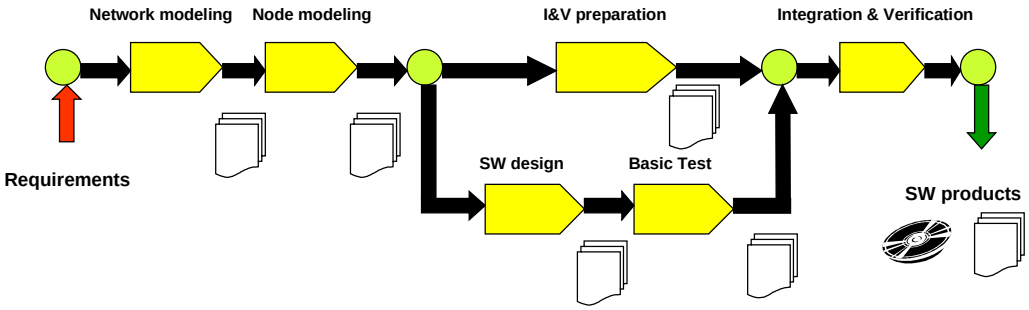
SW modeling
System Structure

- System Structure is not same as the System Architecture, it can be a part of it

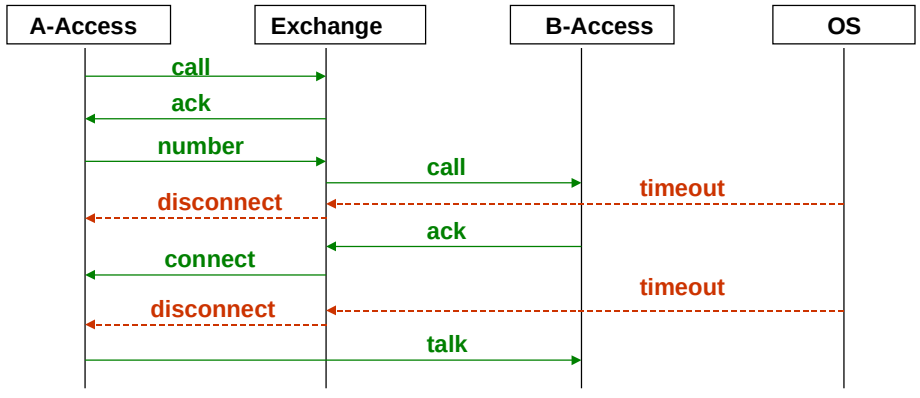


SW modeling
SW development process

- SW development process describes your work flow
- SW development process is a sequence of activities
- SW development process is independent of the tools

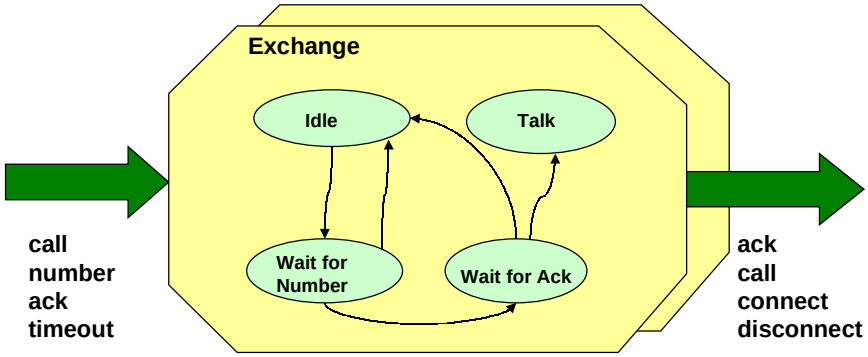


SW modeling
Message Sequence Chart (Call Set-up)



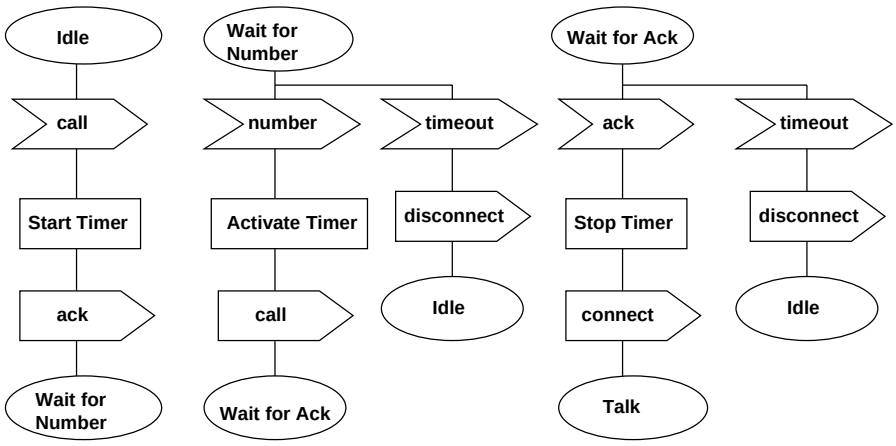
ERICSSON

SW modeling
Finite State Machine (Exchange)



ERICSSON

SW modeling
Flow Chart (Exchange)



ERICSSON

SW modeling
Unified Modeling Language (UML)

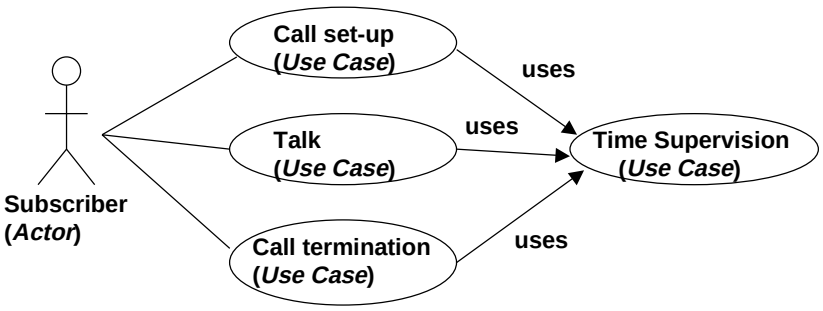
- UML is a object-oriented modeling language for specifying visualizing and documenting the artifacts of an object-oriented system under development.
- As a modeling language the UML allows a description of a system to be made in great detail at any level of abstraction
- It is appropriate for real-time systems
- A model described using the UML syntax can be implemented in any kind of language

ERICSSON

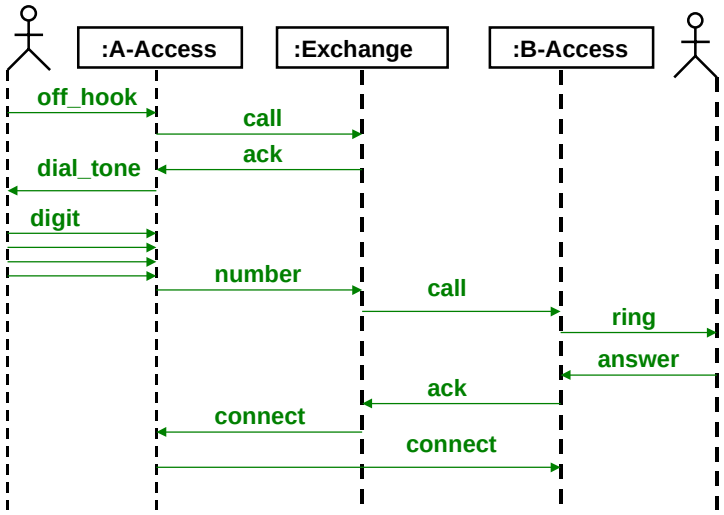
SW modeling
Unified Modeling Language (UML)

- Some UML concepts
  - Use Case Diagram
  - Sequence Diagram
  - Class Diagram
  - Collaboration Diagram
  - Package Diagram
  - State Diagram
- Natural choice for programming language is some object oriented language (e.g. C++, Java, …)
- One UML tool used by Ericsson is “Rational Rose”

SW modeling
UML Use Case Diagram example

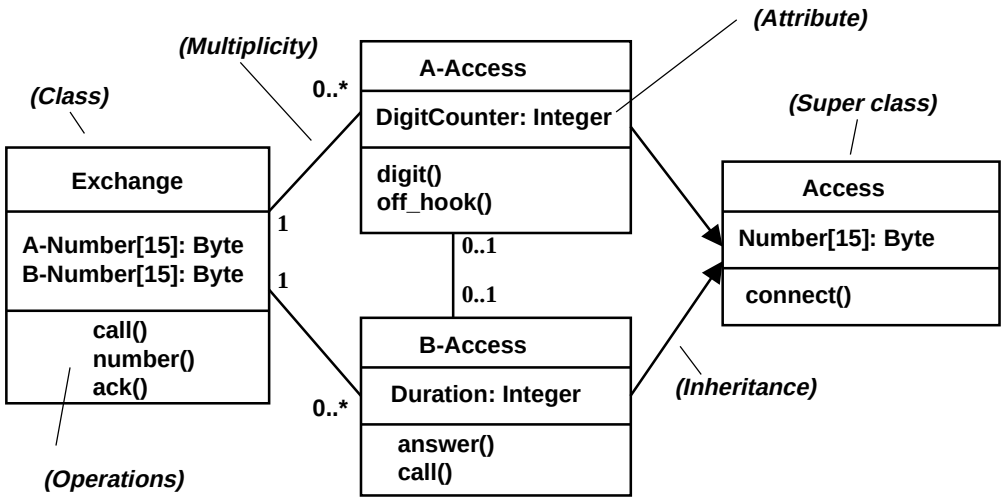


SW modeling
UML Sequence Diagram example (Call Set-up)



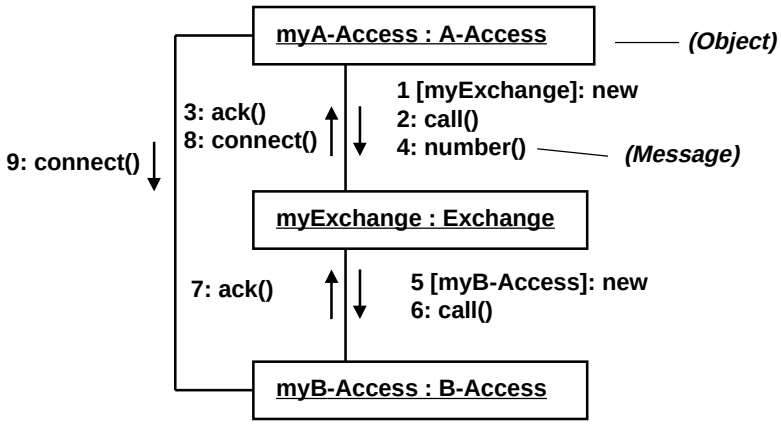
ERICSSON

SW modeling
UML Class Diagram example

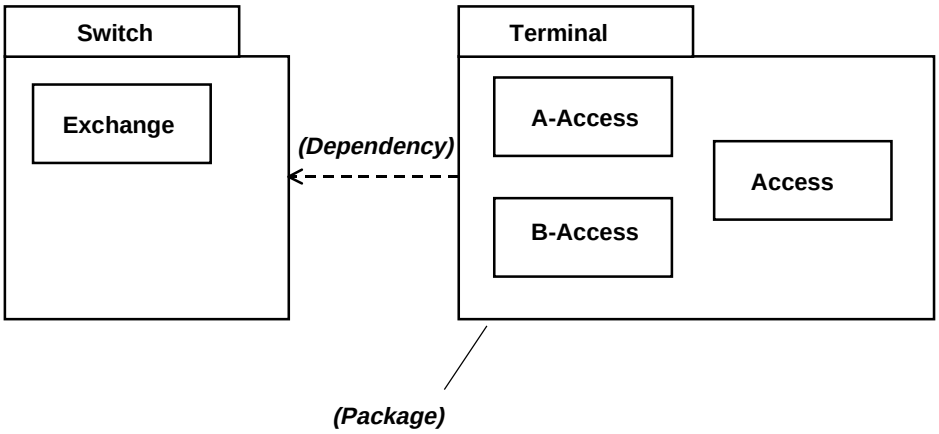


ERICSSON

SW modeling
UML Collaboration Diagram example



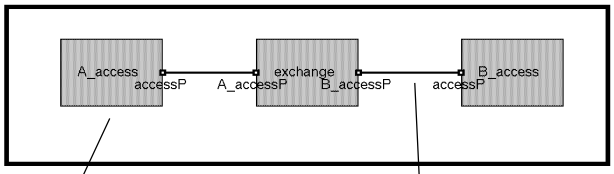
SW modeling
UML Package Diagram example



SW modeling
Real Time Object Oriented Modeling Language (ROOM)

- ROOM is used for specifying, visualizing, documenting and automating the construction of complex, event-driven systems. It is used for higher level abstraction for describing the structure and behavior as a graphical model
- Some ROOM concepts
  - Structure Model
  - Behavior Model
  - Actor
- ObjecTime Developer (OTD) is used by Ericsson for implementing ROOM models
- OTD was established 1992

SW modeling
OTD Structure Model

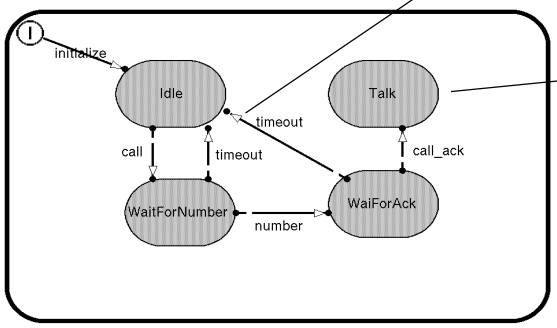


(Actor) *(Protocol)*

SW modeling
OTD Behavior Model (Exchange)

(Transition)

(State)



The transitions can be completed with manual code

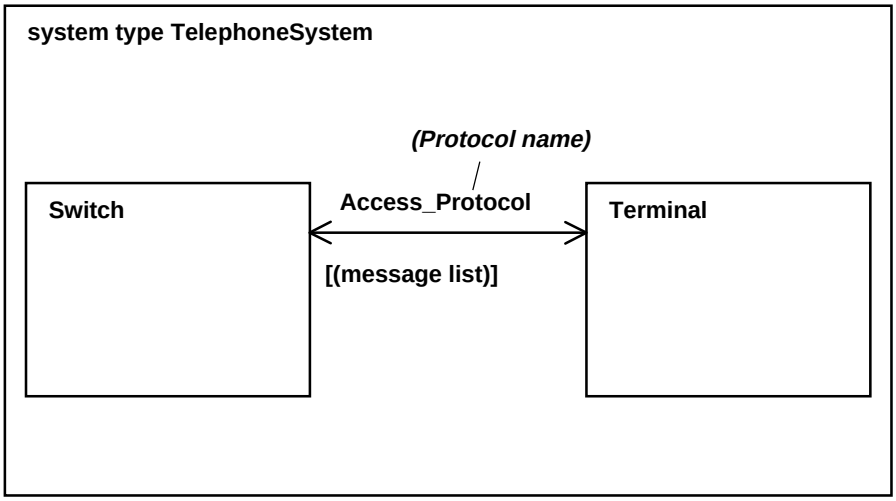
SW modeling
Rational Rose for Real Time

- Rose for RT, the next version of ObjecTime, is powerful tool for visual modeling providing UML-RT for Design, high performance code generation and model execution
- Rose RT provides a tool for the all process phases

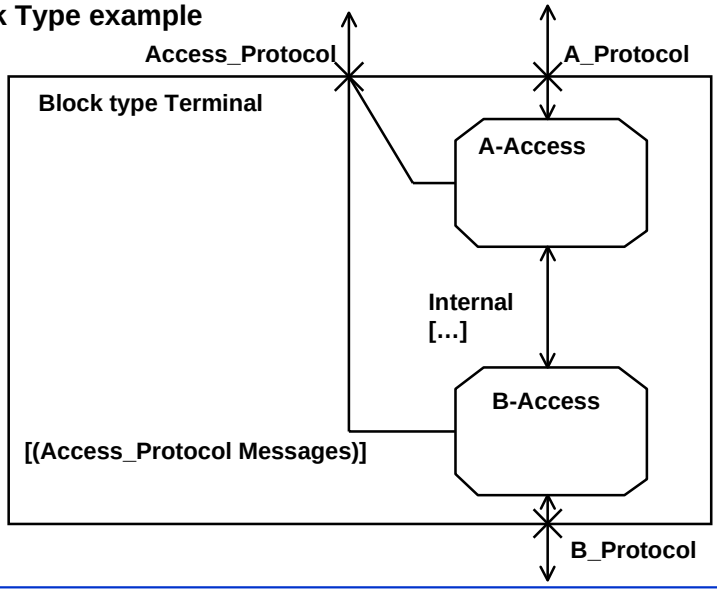
SW modeling
Specification and Description Language (SDL)

- SDL is a standard language for specifying and describing systems
- First version of the language was released 1976
- OO concepts were added 1992
- UML suite 1999 (*Telelogic TAU UML Suite*)
- Some SDL concepts
  - System Type
  - Block Type
  - Process Type
  - Procedure
- Telelogic TAU is used by Ericsson for implementing SDL models

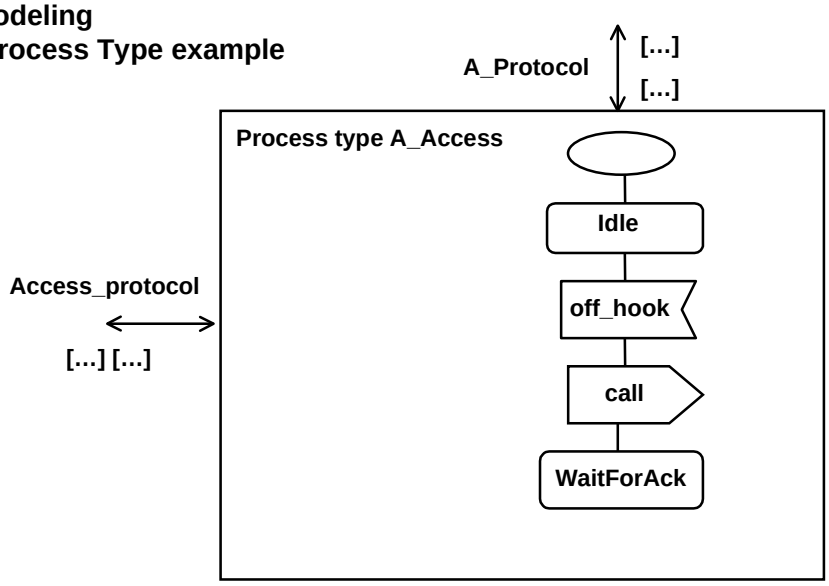
SW modeling
SDL System Type example



SW modeling
SDL Block Type example



SW modeling
SDL Process Type example



Abbreviations

FSM	Finite State Machine
MSC	Message Sequence Chart
OO	Object Oriented
OTD	ObjecTime Developer
ROOM	Real-Time Object Oriented Modeling language
Rose	UML tool from Rational
Rose RT	New version of Rose (Rose Real Time)
SDL	Specification and Description Language
UML	Unified Modeling Language