

RCIS 2012
Sixth International Conference on Research Challenges in Information Science
Valencia, Spain

Tutorial #1

Model-driven requirements engineering in practice

Friday, May 18th 10:30-12:00

At RCIS 2012, Arturo González and Sergio España have offered this tutorial in which a full model-driven engineering method has been explained, trying to highlight many methodological issues that are usually not presented in scientific articles. Also, the industrial experience of Anecoop using Communication Analysis (a communication-oriented requirements engineering method) has been discussed. We would like to thank the tutorial attendees for their time and their questions and reflections.



Model-driven requirements engineering in practice

Arturo González, Sergio España
Centro de Investigación ProS
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

UNIVERSIDAD POLITÉCNICA DE VALENCIA

PRO S
Centro de Investigación en Métodos de Producción de Software

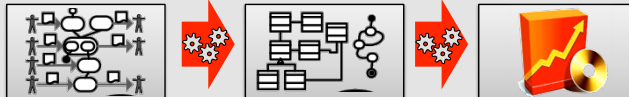
Model-driven development

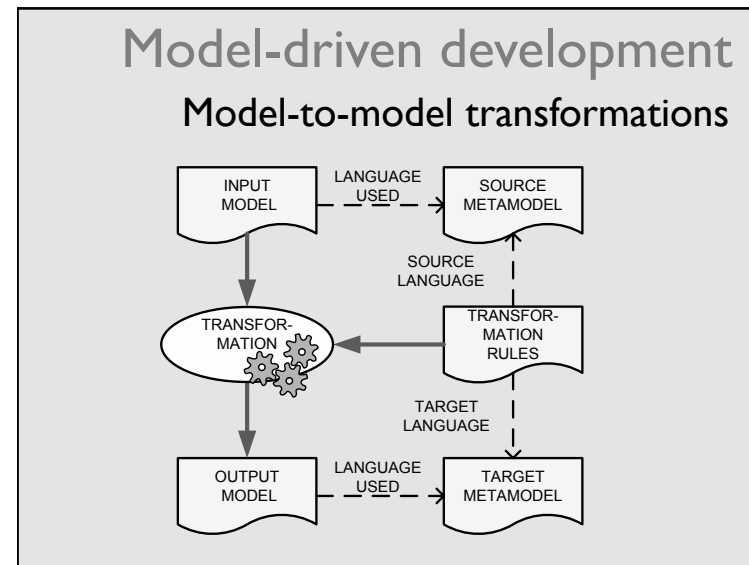
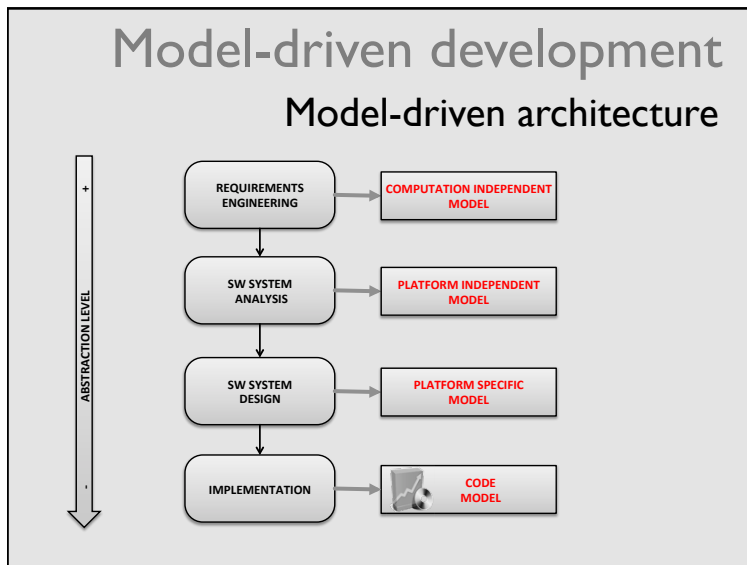
Fundamentals of model-driven development for information systems

UNIVERSIDAD POLITÉCNICA DE VALENCIA

This material is the result of the joint effort of Sergio España, Arturo González, Óscar Pastor and Marcela Ruiz

Model-driven development

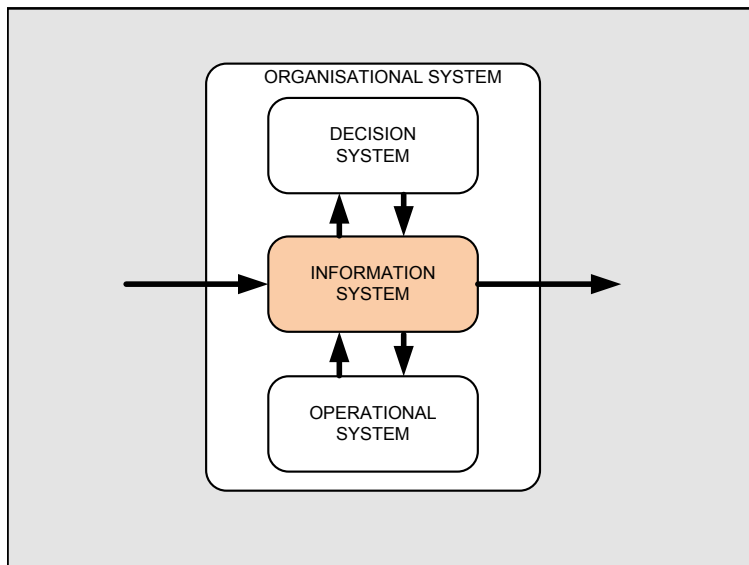




- ## Model-driven development Tool support
- | | |
|---|--|
| <p>AADL from Carnegie-Mellon Software Engineering Institute</p> <p>Acceleo an open source code generator from Obeo</p> <p>Actifsource</p> <p>Apollo for Eclipse from Genteware</p> <p>AndroMDA an open source MDA tool</p> <p>ArcStyler from Interactive Objects Software GmbH</p> <p>Artisan Studio from Artisan Software Tools</p> <p>ASCET from ETAS</p> <p>AtomWeaver from Isomeris</p> <p>CoCoViLa from Tallinn University of Technology</p> <p>EnterpriseCoreObjects by CapableObjects.com</p> <p>Eclipse Modeling Framework (EMF)</p> <p>Enterprise Architect from Sparx Systems</p> <p>GenerateXY from DotXY</p> <p>Generic Eclipse Modeling System (GEMS)</p> <p>GeneXus</p> | <p>Graphical Modeling Framework (GMF)</p> <p>HyperSenses and ANGIE from DELTA Software Technology</p> <p>Innovator from MID GmbH</p> <p>Integranova from CARE Technologies</p> <p>LEONARDI</p> <p>MagicDraw from No Magic Inc</p> <p>ManyDesigns Portofino</p> <p>MetaEdit+ from MetaCase</p> <p>objectiF from microTOOL</p> <p>openArchitectureWare</p> <p>OptimalJ from Compuware</p> <p>Rhapsody from IBM</p> <p>RISE Editor from RISE to Bloom Software</p> <p>SCADE Suite from Esterel Technologies</p> <p>Sculpture Platform from Modelingsoft</p> <p>Select Architect from Select Business Solutions</p> <p>Simulink from MathWorks</p> <p>Together Architect from Borland</p> <p>TOPCASED</p> <p>Uniface from Compuware</p> |
|---|--|


Model-driven development

Think of artefacts that can be engineered...

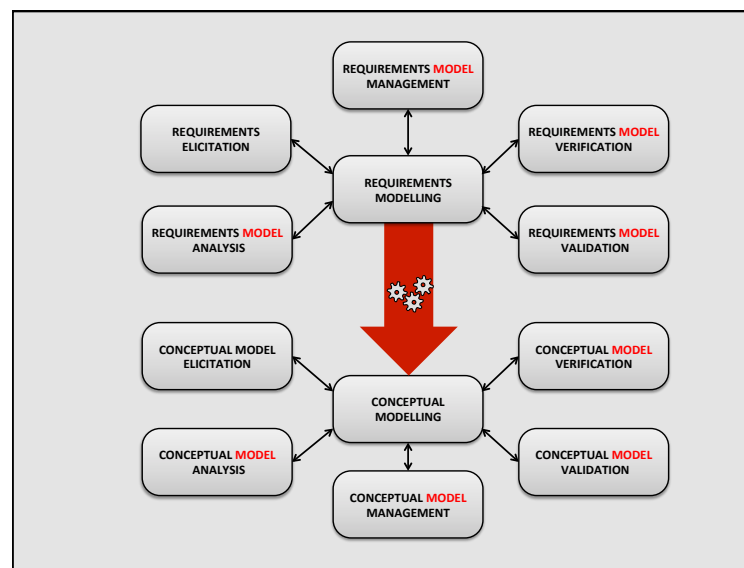
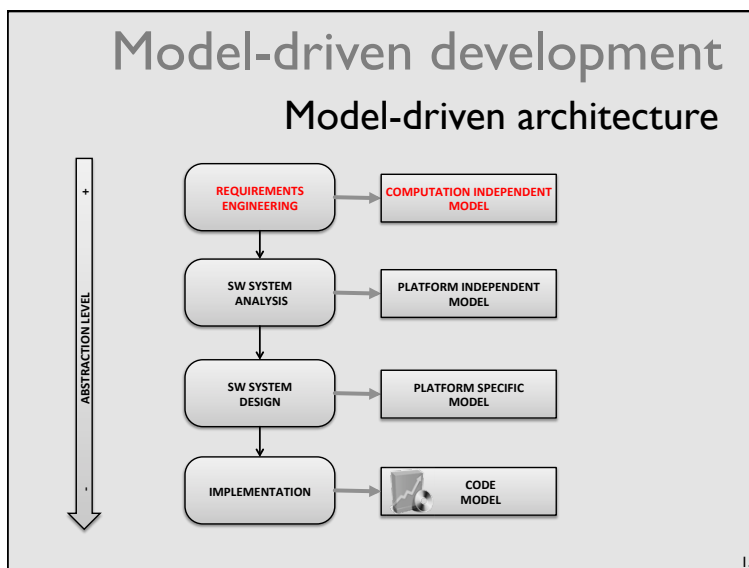


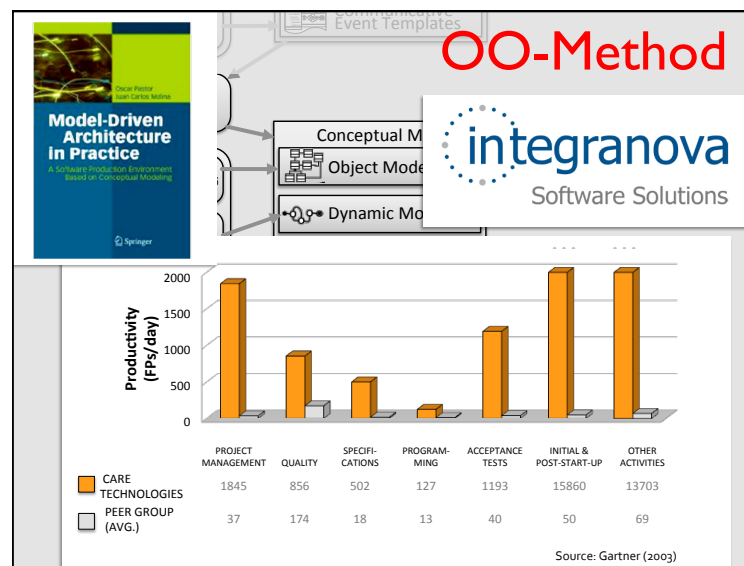
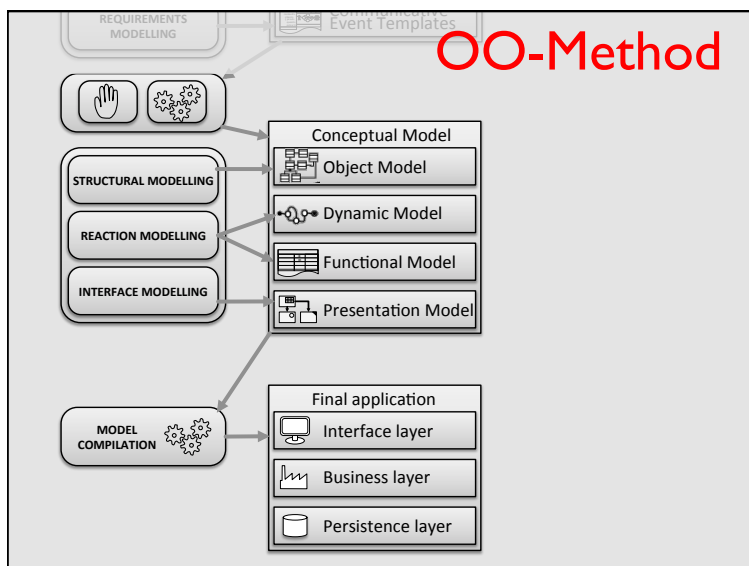
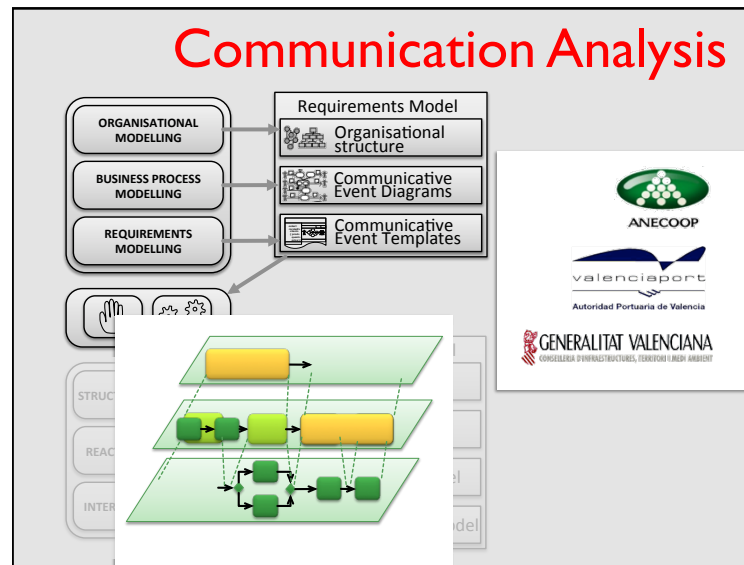
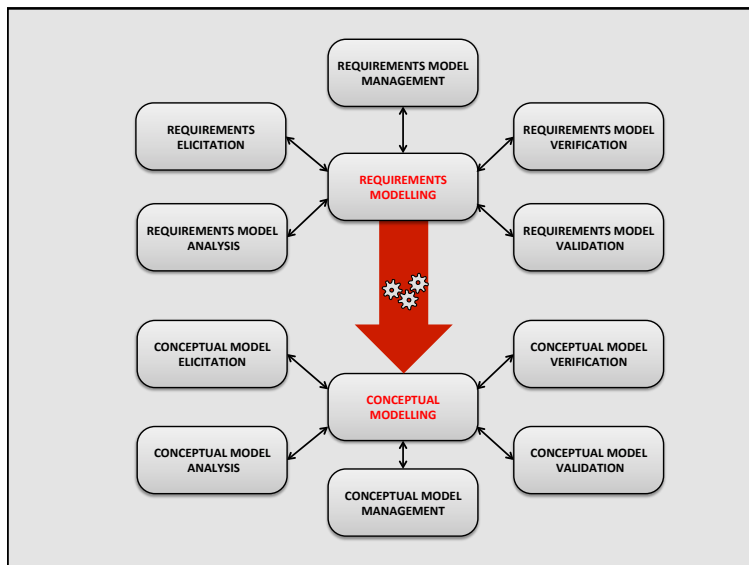
Model-driven development

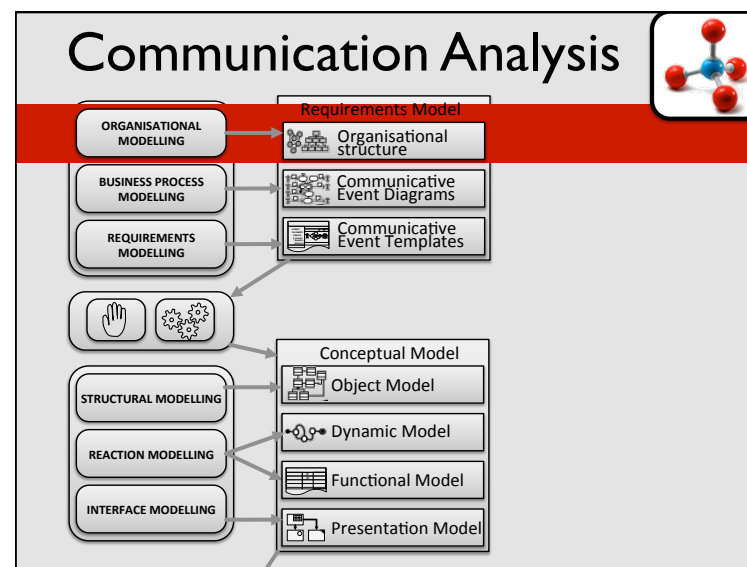
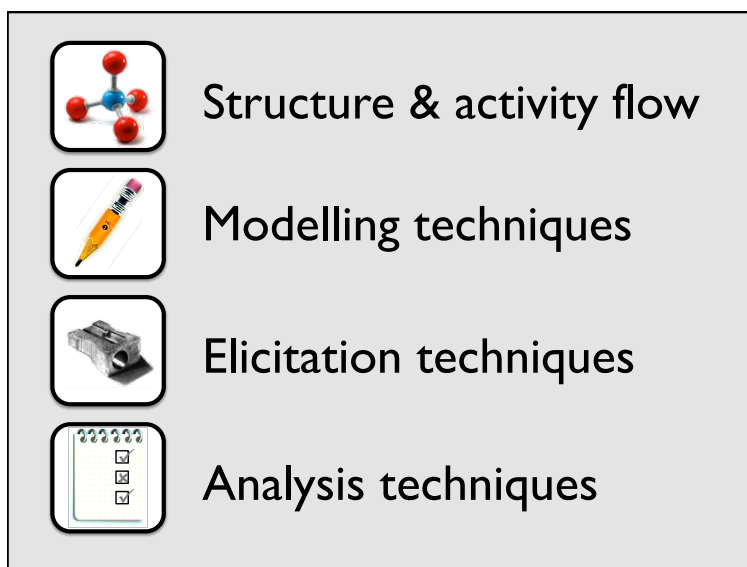
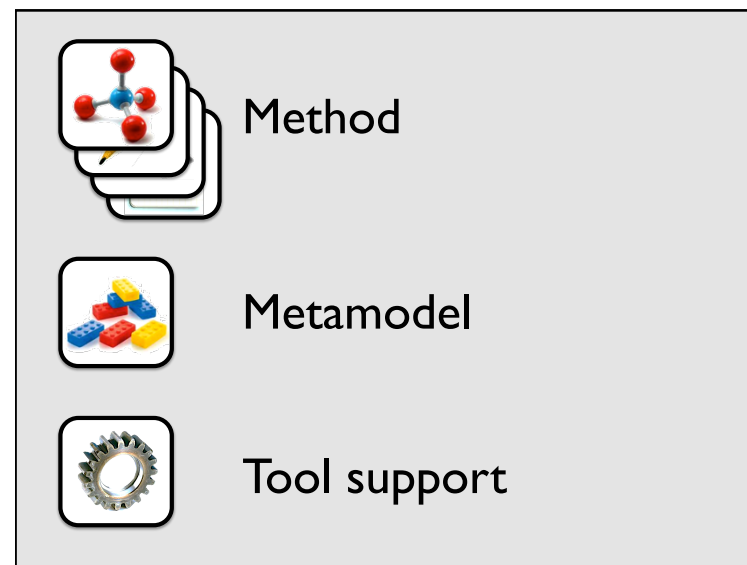
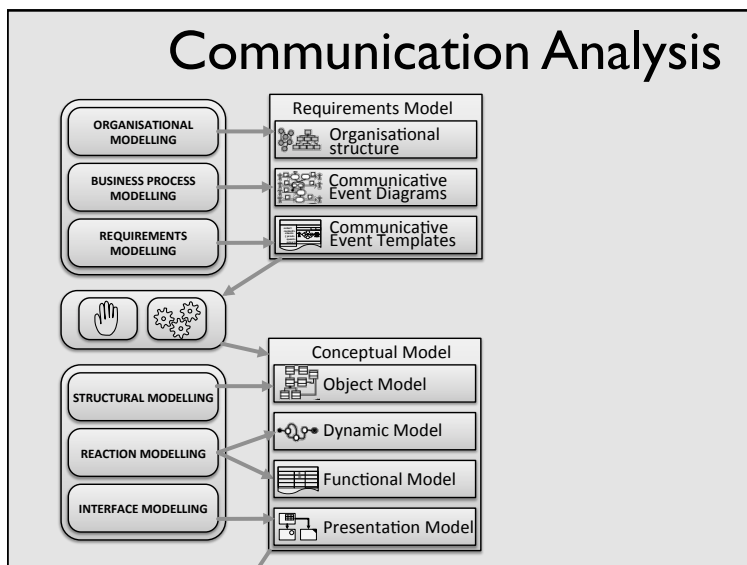
Business process modelling and requirements specification

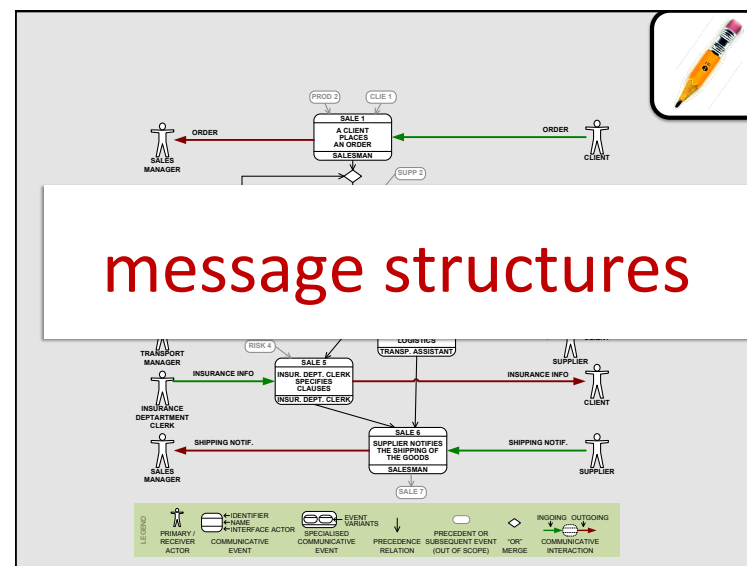
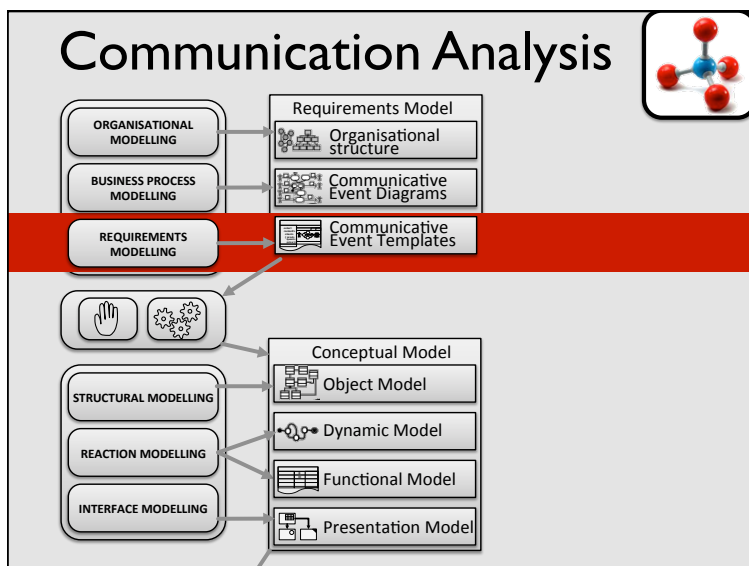
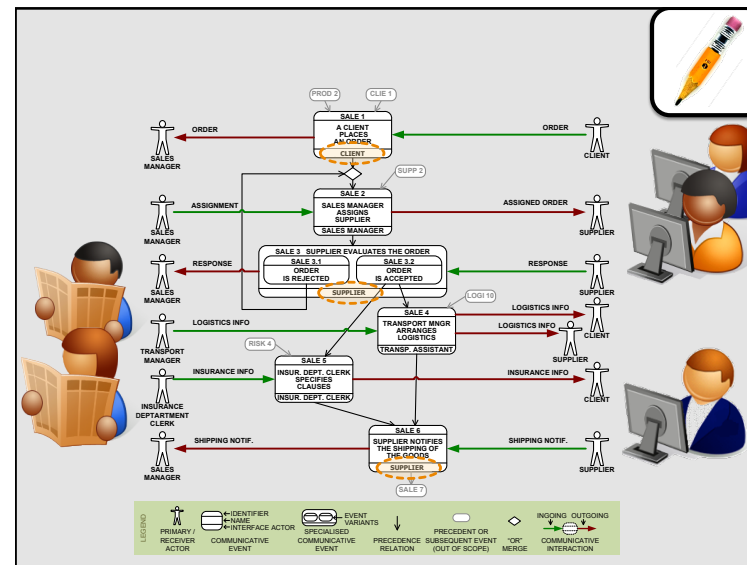
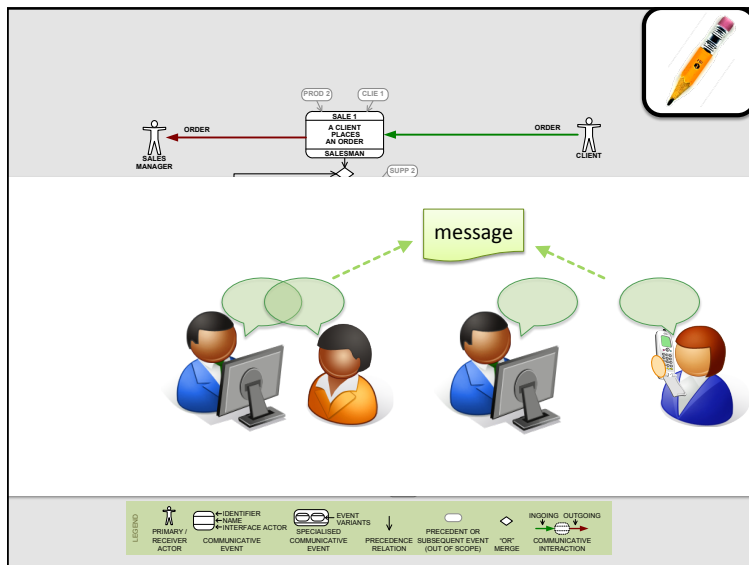

 UNIVERSIDAD POLITÉCNICA DE VALÈNCIA

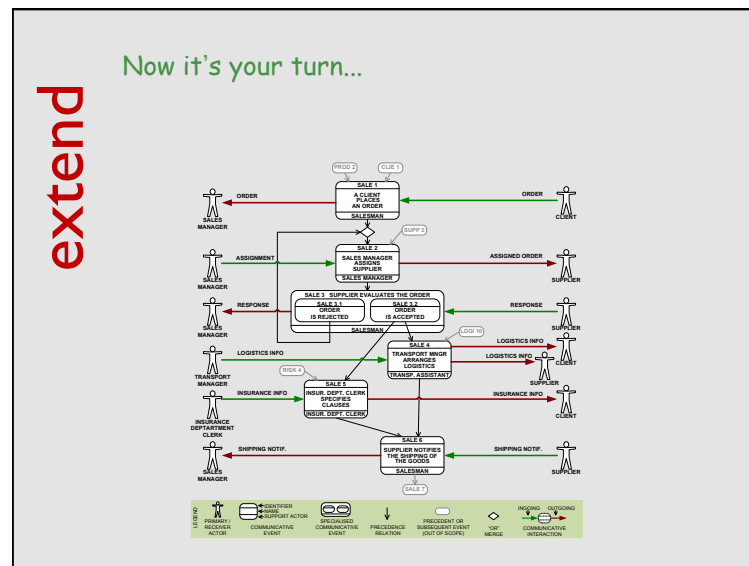
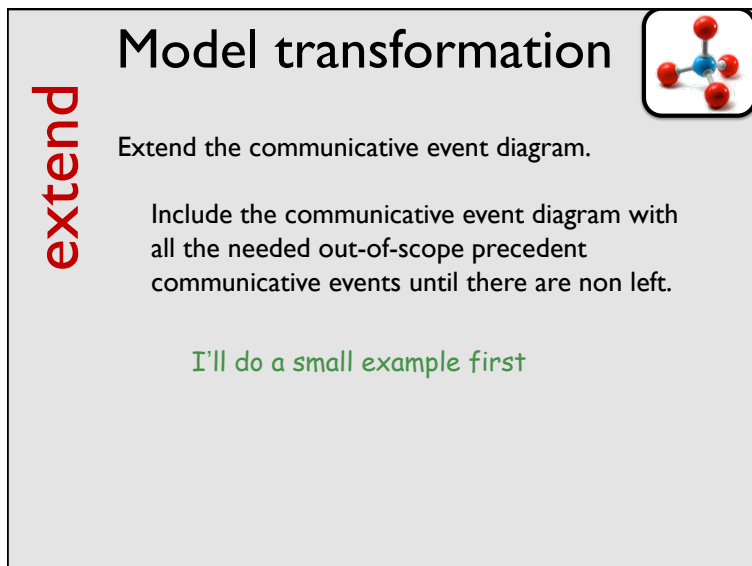
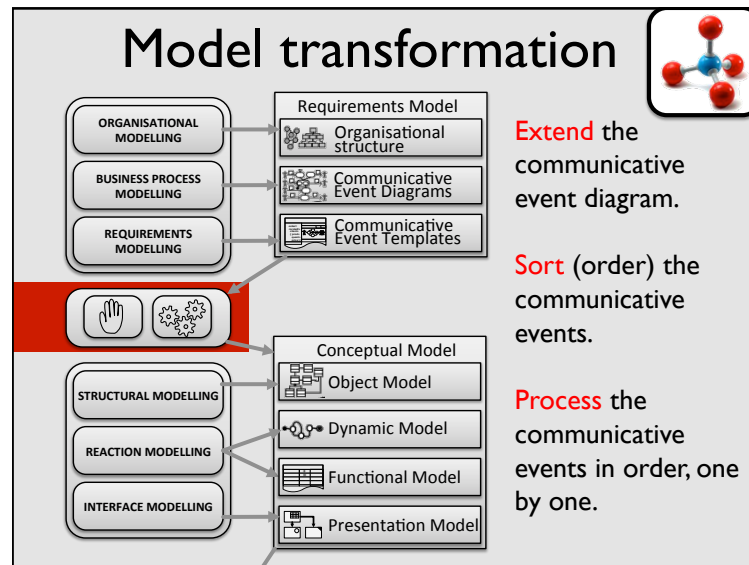
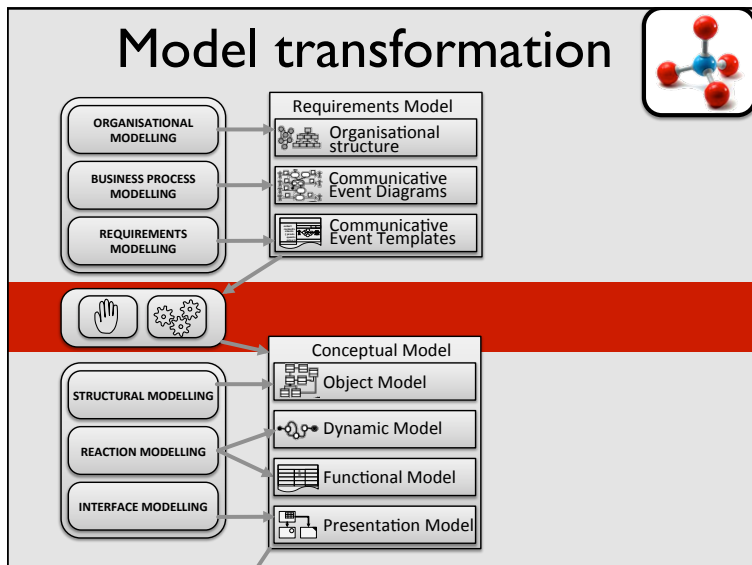
This material is the result of the joint effort of Sergio España, Arturo González, Óscar Pastor and Marcela Ruiz

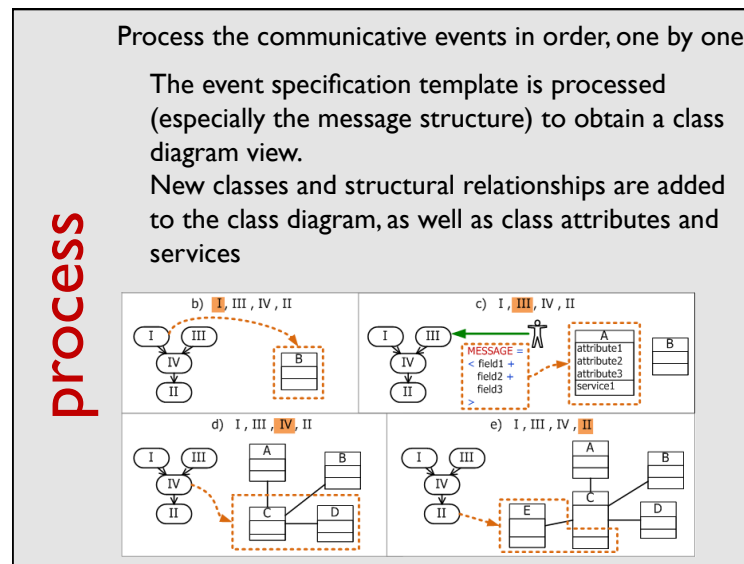
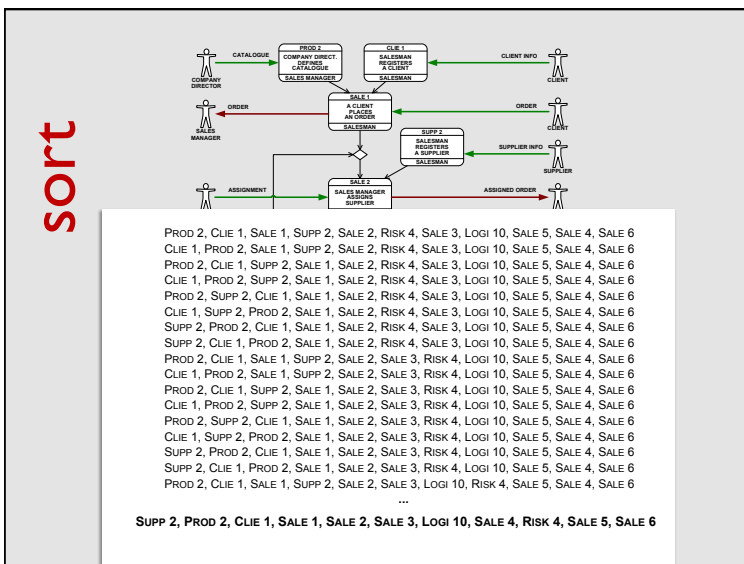
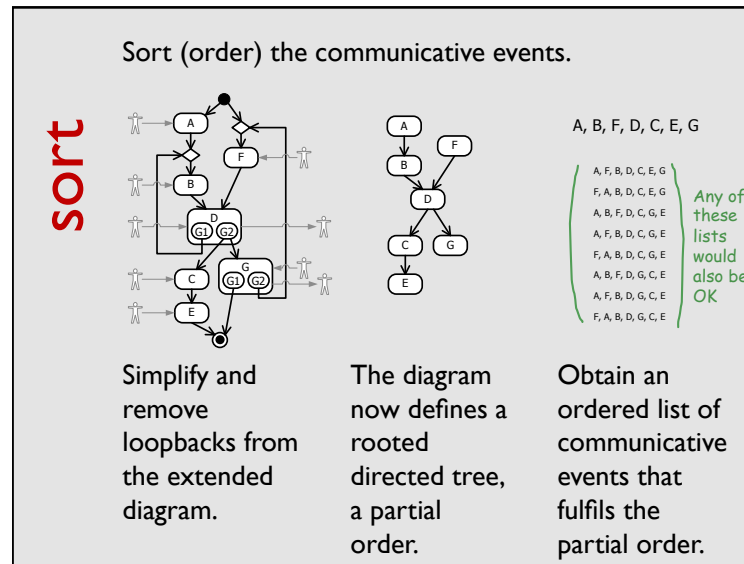
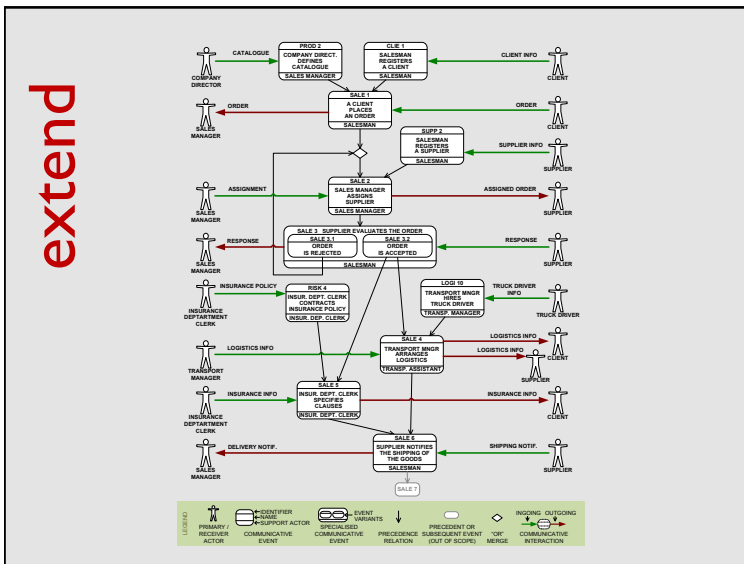












process

Small example

Attribute name	Id	Attribute type	Data type	Size	Requested	Null allowed
supplier_code	yes	Constant	Autonumeric		yes	no
supplier_name	no	Constant	String	120	yes	no
VAT_number	no	Variable	Nat		yes	no
telephone	no	Variable	String	12	yes	yes
address	no	Variable	String	300	yes	yes
post_code	no	Variable	String	12	yes	yes
city	no	Variable	String	100	yes	yes
registration_date	no	Variable	Date		yes	yes

process

Class

Attributes | Derivations | [Attribute value expressions] | Services | Transactions | Constraints | Agents | Relationships | General

Name	Attribute type	Data type	Id	Size	Default v...	Request ...	Nulls
supplier_code	Constant	Autonumeric	Yes			Yes	No
supplier_name	Variable	String	No	120		Yes	No
VAT_number	Variable	Nat	No			Yes	No
telephone	Variable	String	No	12		Yes	Yes
address	Variable	String	No	300		Yes	Yes
post_code	Variable	String	No	12		Yes	Yes
city	Variable	String	No	100		Yes	Yes
registration_date	Variable	Date	No			Yes	Yes

Name: Attribute type:

Alias: Data type:

Help Message:

Comments:

Class:

process

Small example

Argument name	Data type	Size	Null allowed
p_atrsupplier_code	Autonumeric		no
p_atrsupplier_name	String	120	no
p_atrVAT_number	Nat		no
p_atrtelephone	String	20	yes
p_atraddress	String	300	yes
p_atrpost_code	String	12	yes
p_atrcity	String	100	yes
p_atrregistration_date	Date		yes

process

Class

Attributes | Derivations | [Value expressions] | Services | Transactions | Constraints | Agents | Relationships | Alternate Id | General

Events & Transactions

Name	Features
Supp2_new_supplier	New

Inbound Arg	Outbound Arg	Name	Data type
		p_atrsupplier_code	Autonumeric
		p_atrsupplier_name	String
		p_atrVAT_number	Nat
		p_atrtelephone	String
		p_atraddress	String
		p_atrpost_code	String
		p_atrcity	String
		p_atrregistration_date	Date

Event: Transaction: Operation:

New Internal use Destroy

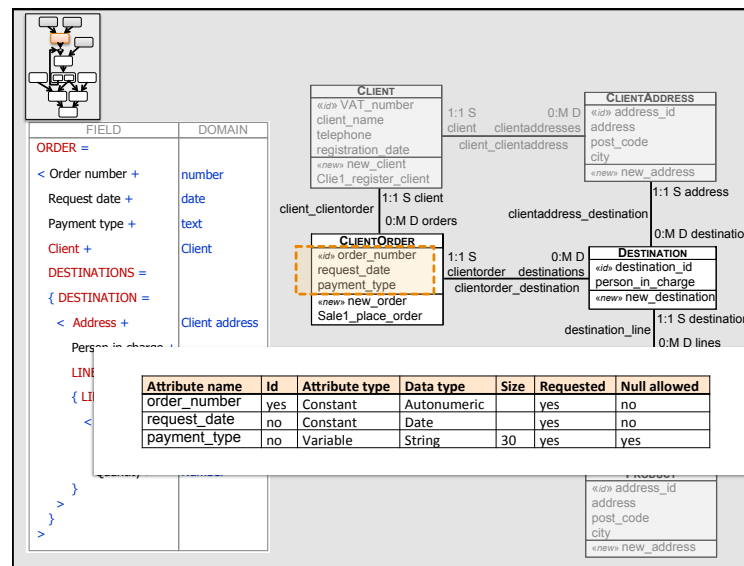
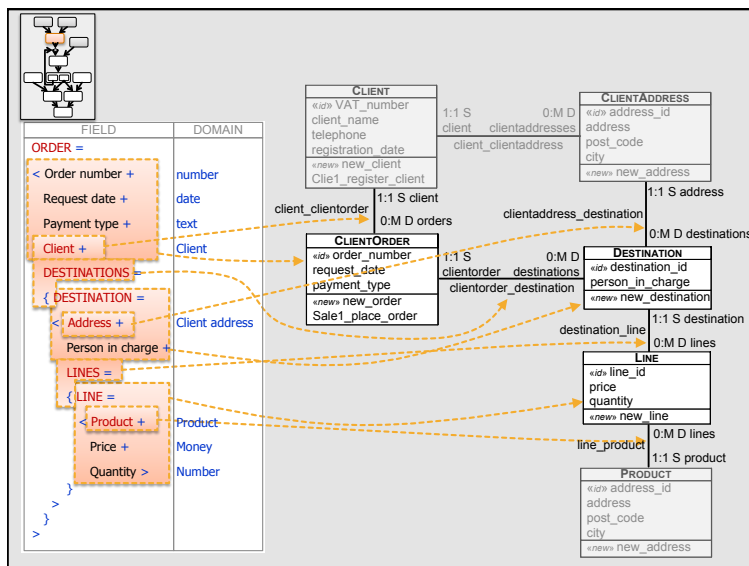
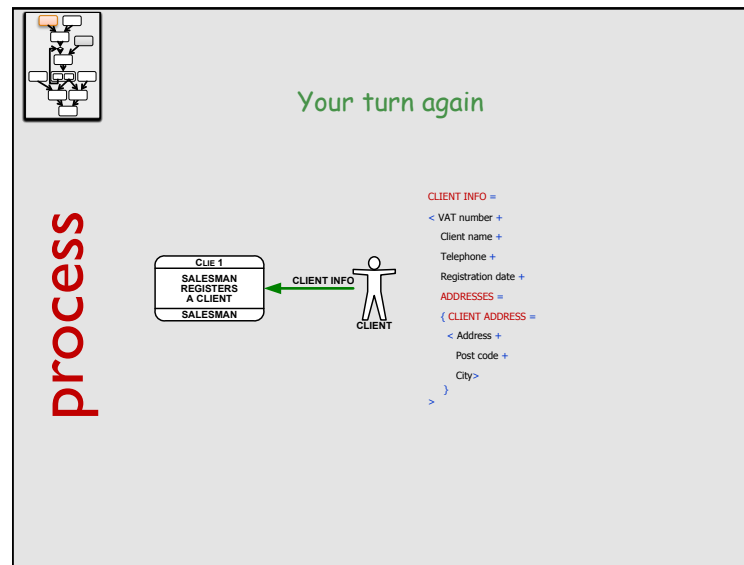
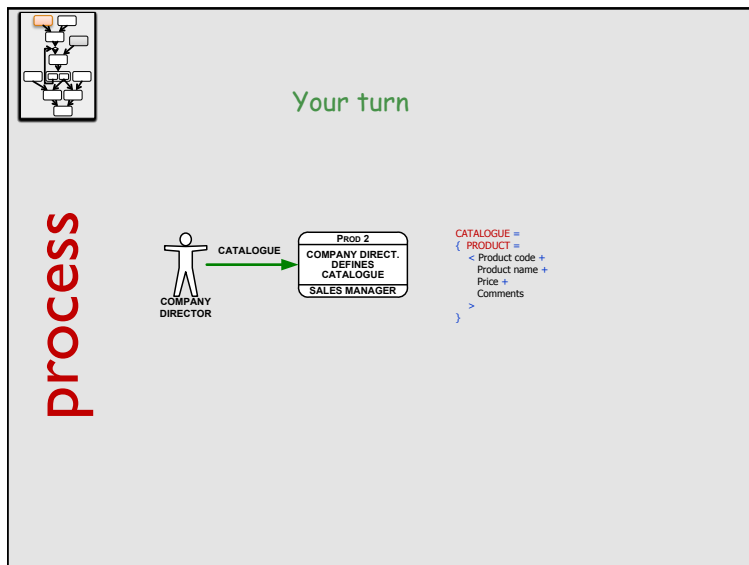
Name:

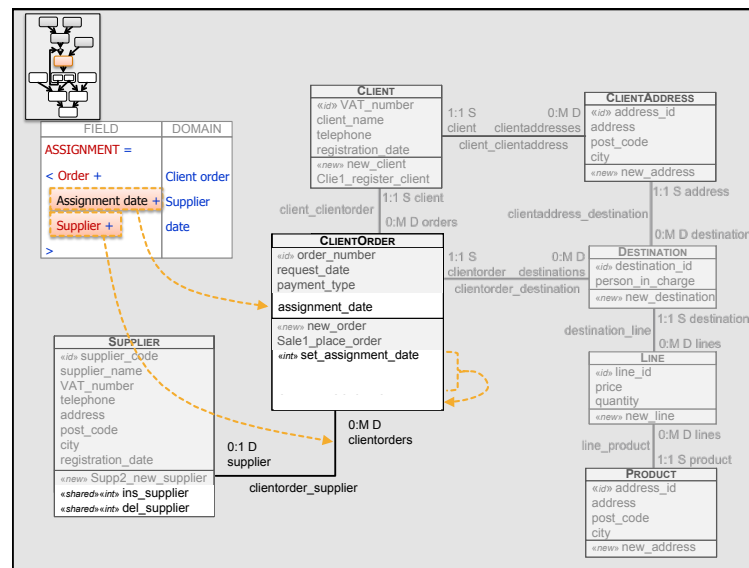
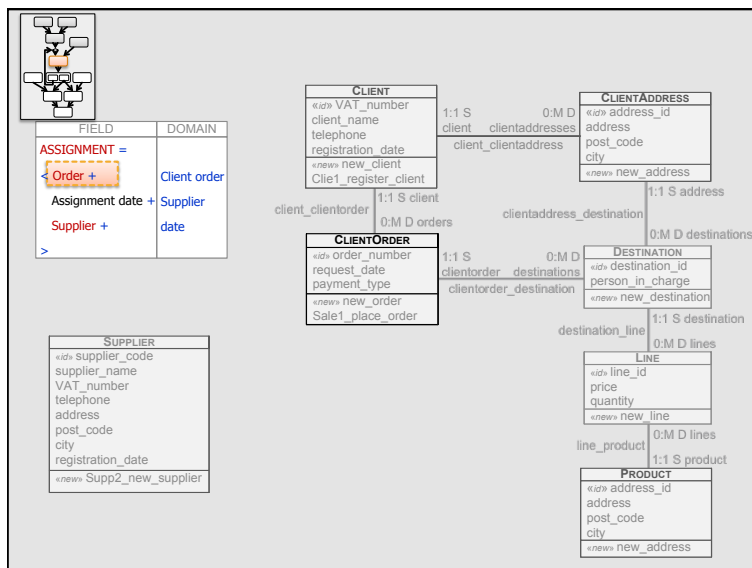
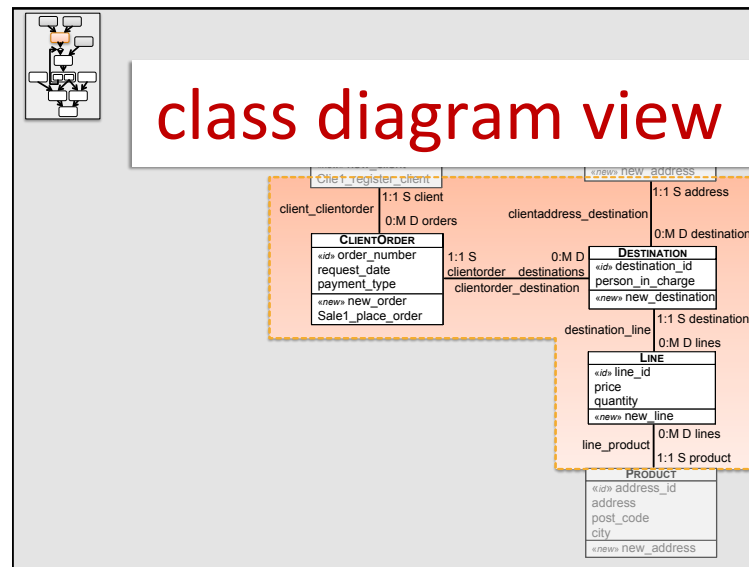
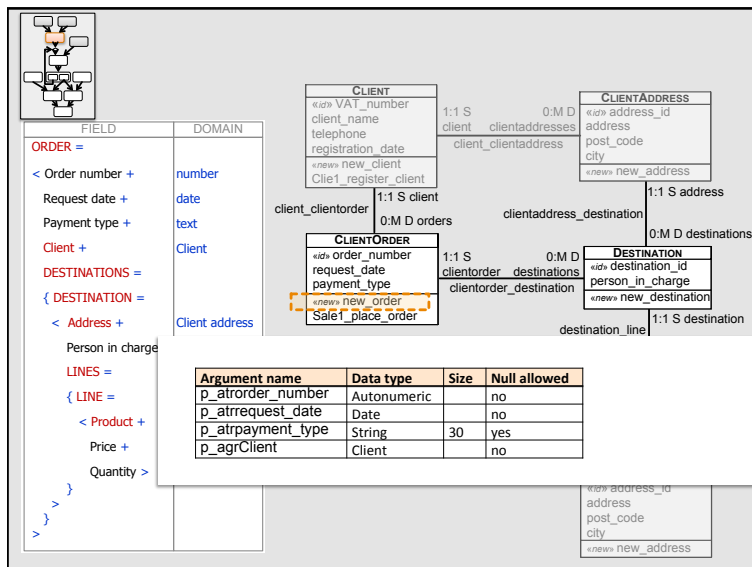
Alias:

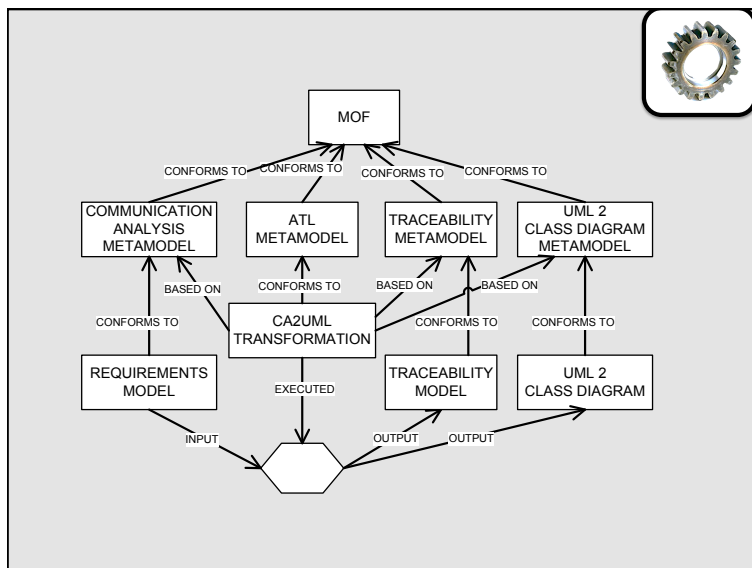
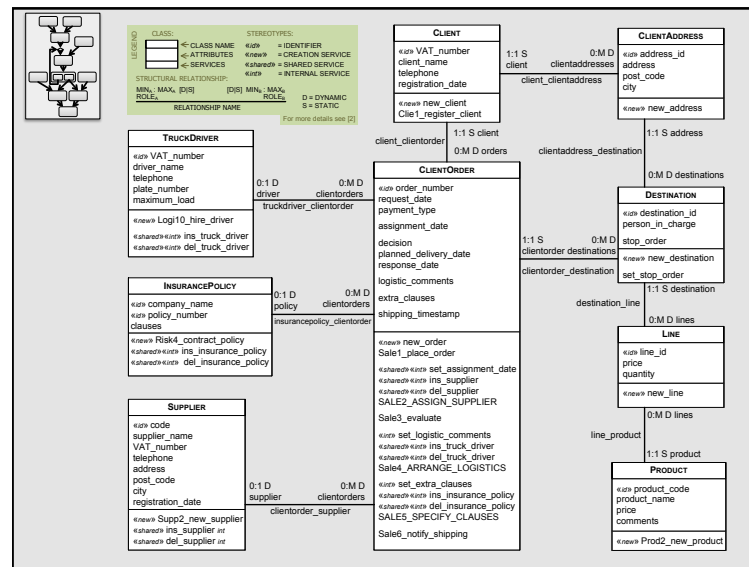
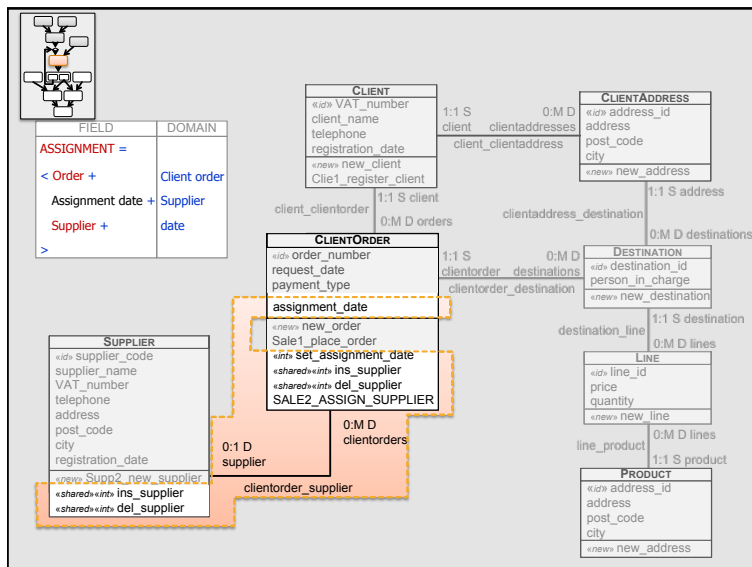
Help message:

Comments:

Class:

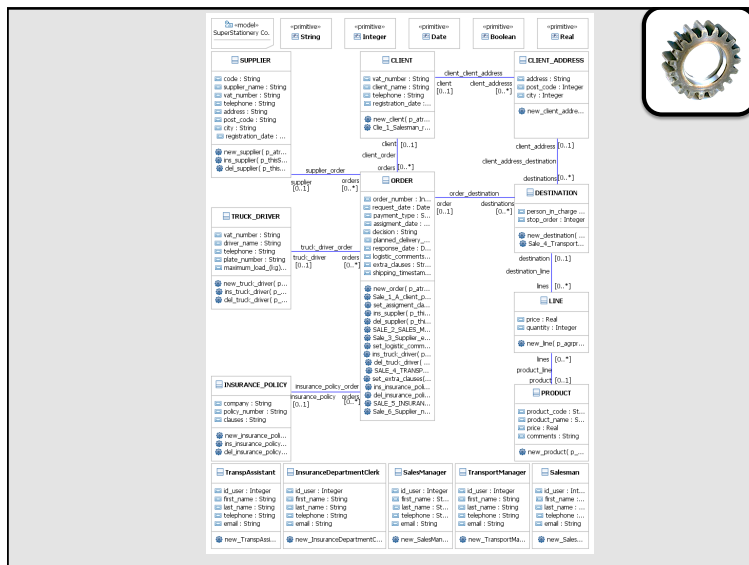







```

    rule process_create_class(active_substructure: cametamodelSubstructure) {
    to
    class: umlClass(name<-self.format_to_undercore(self.format_to_trim(active_substructure.name))),
    elementSrc: traceElement(elementName<- active_substructure.name, elementType <- "AggregationSubstructure"),
    elementTrg: traceElement(elementName<- class.name,elementType <- "Class",targetTrace<-Sequence(elementTrg)),
    trace: traceTrace(
        traceName<- "Class_Derivation",
        targetElement <- elementTrg)
    do {
    --Establish active class
    self.active_class <- class;
    --Traceability
    elementSrc.refSetValue(sourceTrace, Sequence(elementSrc));
    trace.refSetValue(sourceElement, elementSrc);
    --Set tuples substructure and class
    self.sub_classes_set.add(self.sub_subclass(active_substructure, self.active_class));
    --Add the new class to the Model
    self.conceptual_model.packageElement.add(self.active_class);
    }
    }
    
```

Model-driven development

Do not hesitate to write us for additional information, to provide some feedback or to propose a collaboration.



Arturo González, Sergio España
 agdelrio@dsic.upv.es
 sergio.espana@pros.upv.es

Please allow us an advertisement :)

CAiSE'12

GDAŃSK, POLAND 24th International Conference on Advanced Information Systems Engineering

Tutorial 2

Full model-driven practice: from requirements to code generation

Thursday, June 28th 14:30-16:00

At CAiSE'12, Óscar Pastor and Sergio España will be offering another tutorial that complements this one at RCIS 2012. Although we will review the whole MDD process (from requirement engineering onwards), we will offer insights in the model compilation; that is, the automatic generation of fully-functional software code from platform-independent conceptual models. In that case, the industrial experience that will be explained is that of CARE technologies, creators of the *Integranova Model Execution System*.