

1. ABSTRACT

A crucial success factor in information systems development is the alignment of the system with business goals, business semantics and business processes. Developers should be freed from programming concerns and be able to concentrate on these alignment problems. Model-driven system development (MDD) not only provides a structured and systematic approach to systems development, but also offers developers the possibility of using model-transformation technologies to derive models of a lower abstraction level that can be further refined, and even generate software code automatically.

This tutorial will show how to successfully integrate business process modelling (BPM), requirements engineering (RE) and object-oriented conceptual modelling with the objective of leveraging MDD capabilities. Participants will be presented state of the art modelling methods and code generation tools to explore different ways to match an information system with business requirements.

The tutorial presents the principles, concepts and common practices of MDD, with a special focus on **model-driven requirements engineering**.

- We will offer insights on how to elicit and specify the requirements of an information system, including the following abilities:
 - To ask the proper questions in order to discover and disambiguate user needs.
 - To structure and organize appropriately the set of requirements into a business process model and a requirements model.
- it will be discussed how to create the object-oriented conceptual model of the computerised information system, including the following abilities:
 - To systematically derive an initial conceptual model from the requirements model.
 - To complete the conceptual model in order to specify the software system considering both its static and dynamic aspects.

Also, it will be shown how to manage the necessary tools to support a model-driven development that covers the whole lifecycle: from RE to software-code generation. These include general purpose diagramming tools and conceptual modelling tools, as well as a state-of-the-art model compiler that automatically generates fully-functional source code from the conceptual model.

In short, this tutorial offers a broad view on model-driven RE; that is, how business process models and requirements models can be embedded in a complete MDD process. As a practical application a specific method and notations are explained, but the ultimate goal is that participants are able to apply this knowledge to their own contexts, to either industrial practice or academic research.

2. INTENDED AUDIENCE

In view of recent advances of MDD tools, it seems likely that, in a near future, many full-fledged CASE tools will support **model-driven requirements engineering**. This tutorial is targeted at any stakeholder that wants to anticipate to that situation, including (but not restricted to):

- **Researchers** who want to grasp an integrated view of the whole MDD paradigm; the tutorial will quickly review the state of the art, offer a successfully industry-adopted RE method coming from academia, and highlight future research challenges in the area.
- **Practitioners** who want to know how to fit RE and MDD pieces together in a way that will work in practice; the tutorial will offer guidelines and best practices that you can later tailor to suit your own methods and notations.

We will use a running example to clarify concepts.

3. PRESENTERS

Arturo González, PhD, is full professor and at the Universitat Politècnica de València (UPV, Spain). He has large experience in complex software development projects as requirements engineer and consultant. He has successfully transferred Communication Analysis, a communication-oriented business process modelling and RE method to industry by applying the method in his latest projects for the Valencia Port Authority, for the Infrastructure and Transport Ministry of the Valencian Regional Government, and for Anecoop S. Coop. (the biggest fruit and vegetables second-grade cooperative in Spain). He has lately been involved in the integration of RE methods into MDD frameworks. He has been the director of the Master in Information Systems Management at UPV for more than 10 years.

agdelrio@dsic.upv.es

Sergio España, PhD, is a research fellow at the Research Center on Software Production Methods (ProS), Universitat Politècnica de València (Spain). He has published in top RE and conceptual modelling conferences. He is member of the programme committee of several international workshops (e.g. ONTOSE, VORTE). His main research interests include information systems, conceptual modelling, RE, interface design, empirical software engineering and model transformations. He is co-author of Communication Analysis, an information systems RE method that can be applied stand-alone or within an MDD framework. He leads the RE research line in ProS.

sergio.espana@pros.upv.es

4. OUTLINE OF TOPICS

The tutorial will cover the following topics:

- **Quick introduction to MDD**, the Model-Driven Architecture and model transformations. Underlying concepts and state of the art in methods and technology.
- **Requirements engineering**. How to use BPM and RE to capture organisational needs in a concise yet expressive way, focusing in organisational communications. Not only it is important to describe requirements, but to have the methodological tools to discover and agree them successfully. How to appropriately structure requirements and focus on those that will allow for model transformations.
- **Conceptual modelling**.
 - **Using requirements in an MDD way**. How to systematically derive an initial version of an object-oriented conceptual model from the business process models and the requirements models.
 - **Further conceptual model refinement**. How to complete this model in a way that it specifies the structural, dynamic and interaction views of the system.

As a practical application, the OO-Method a conceptual modelling framework with industrial tool support will be explained. It includes the following complementary views: Object Model (a Class Diagram that structures system memory); Dynamic Model (a set of State Machines that model valid object life-cycles); Functional Model (an abstract pseudocode that specifies the reaction of objects to events); Presentation Model (an abstract user interface model). A demo of automatic code generation for different platforms will be made.

Participants will be introduced to using several tools to effectively support model-driven RE, ranging from general-purpose (e.g. MS Visio, MS Word) to CASE tools (e.g. *OLIVANOVA* Modeler <http://www.care-t.com>)

The material of the tutorial is mainly the following:

- Slides and extended textual material that expound the subjects of the tutorial.
- Links to related material and tools, for those participants that want to know more about the topics.

5. MOTIVATION

MDD is, no doubt, an active area of research and innovation nowadays. MDD first covered the conceptual-modelling stage, paving the way to industrial tools that support modelling and code generation. Now, approaches that adapt RE methods to the MDD paradigm (or even invent new ones) start to appear, along with model-transformation strategies to derive conceptual models from requirements models. It is a matter of time that many methods and tools are available for model-driven RE. However, for the present, not many of them are mature enough to be applied under conditions of practice.

This tutorial offers a vision of the state of the art and it will present a systematic approach to model-driven RE. More importantly, although the tutorial will introduce participants to a cutting-edge technology, the main focus is put on methodological guidelines: it intends to provide insights on the topic that are useful to both researchers (so they can apply them in their proposals) and practitioners (so they are aware of what is coming and can anticipate the evolution of RE methods and CASE tools).

SAMPLE MATERIAL

We have prepared material for explaining the concepts and training in the methods, examples to illustrate the proposals, exercises to increase the participants' competence in model-driven requirements engineering.

MDRE
RE'2011

Information systems

What does an information system do?

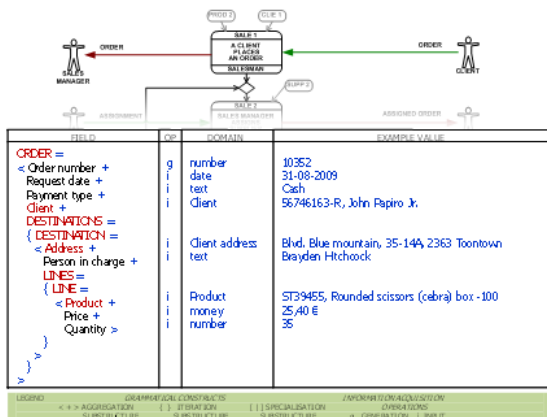
- An IS deploys a observation network that aims to find out about its environment:
 - the events that occur
 - the state of things
- The information system consists of...
 - people (e.g. clerks, archivists)
 - machines (e.g. sensors, computers, calculator)
 - software (e.g. sales management application, email server)
 - other resources (e.g. paper forms, pencils, filing cabinets)
 - norms (e.g. processes, protocols)
- Each time an event occurs, the IS has to ensure that a descriptive message is received.
- To carry out their tasks, the members of the organisational system need to consult the IS, in order to be aware of (or remember)...
 - what has happened before
 - in which state things are



The slides are used to define concepts, illustrate their meaning and present the methods and techniques.

They are intended to foster participation and debate.

Requirements engineering



4. USAGES OF MESSAGE STRUCTURES

Message Structures can be applied for different purposes (from software development to adaptive maintenance) and in different stages of the software development life cycle (e.g. analysis, design). Depending on whether they are used in analysis or design time, syntactic and pragmatic differences have to be taken into account. Table 5 presents recommendations on the usage of field properties, depending on the development stage in which Message Structures are used.

Table 5. Applicability of field properties to development stage

	Name	Acquisition operation	Domain	Example	Description	Label	Link with memory	Computativeness	Initialisation	Visibility
Analysis	++	++	++	++	++	++	++	++	++	++
Design	Memory	++	++	++	++	++	++	++	++	++
	Interface	++	++	++	++	++	++	++	++	++

USING: ++ highly recommended + recommended - not recommended ** discouraged

4.1. Creation and usage of Message Structures in analysis time

In analysis time, Message Structures allow specifying in detail the communicative interactions that take place in the organisational work practice. This way, they offer a communicational perspective for business process modelling and they act as requirements for the IS. In the context of Communication Analysis, the new meaningful information that is conveyed to the IS in each communicative event is specified by means of a message structure.

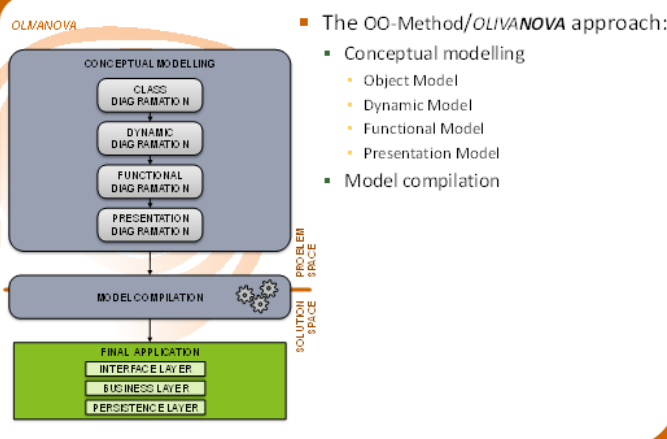
In the following, we enumerate some sources of information and techniques for acquiring information and analysing the messages exchanged with the IS.

Organisational actors play an important role in IS analysis, since they know organisational work practice first-hand. The analyst will employ elicitation techniques such as interviews or JAD sessions [August 1991]. It is crucial to ask the proper questions so as to define which information is conveyed in each communicative event, as well as to distinguish new information from derived information.

Business forms are a technological support for communicative interactions and, therefore, they are a major source for analysis. In this sense, the user interface screens from pre-existing software are equivalent. Forms can be used for entering information (input forms), for presenting data (output forms), or for both purposes. In analysis time, input forms allow to identify communicative events that convey new information to the IS. The analyst has to carry on, along with the users, the following investigations:

- Whether the form is filled in one go or iteratively in different moments in time; the corresponding communicative events are identified. For instance, the client order form shown in Fig. 1 is affected by more than one communicative event? [España, González et al. 2011]: the request of the order, the assignment to a supplier, and the supplier response.
- Which is the temporal order in which communicative events occur.
- Which are the primary actors of the communicative events (those that are the source of the information the form is filled with).
- What message is conveyed in each communicative event (the fields of the form that are affected by the event). Observe that the message structure in Table 1 does not include fields such as Supplier or Planned delivery date, which correspond to later communicative events.

Conceptual modelling

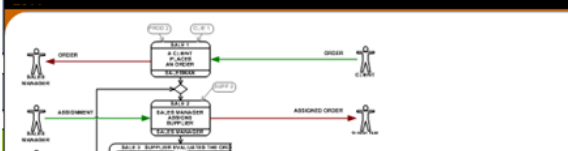


The OO-Method/OLIVANOVA approach:

- Conceptual modelling
 - Object Model
 - Dynamic Model
 - Functional Model
 - Presentation Model
- Model compilation

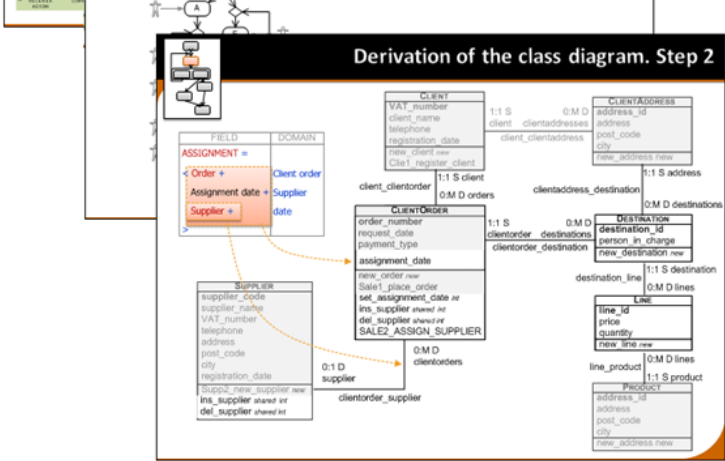
Besides, there is textual material available that provides more information on the commented topics.

Some of the material can be given in advance to registered participants.



Derivation of the class diagram. Pre-processing

- Sort (order) the business activities.
 - Remove any loopback precedence relationships from the extended communicative event diagram.
 - The diagram now defines a rooted directed tree → a partial order.
 - Obtain an ordered list of communicative events that fulfils the partial order.
- I'll do a small example first



A running example illustrates how model-driven requirements engineering can be applied in practice. This case consists on a textual description, its corresponding business process models and requirements models, an explanation of the systematic derivation of the conceptual model, and the resulting conceptual model.

4.2.7. SALE 2. Sales Manager assigns supplier

After an order is placed, the Sales Manager assigns the order to one of the many suppliers that work with SuperStationery. Thus, communicative event SALE 1; namely, the client order. Note that the reference field Order in the message structure of SALE 2 indicates the business object being modified. This way, the class that corresponds to this business order is extended; namely, CLIENTORDER is affected by SALE 2. The data fields in the message structure of Sale 2 lead to adding new attributes to this class, whereas the reference fields in the message structure lead to adding new structural relationships between class CLIENTORDER and other classes that (presumably²²) already exist in the class diagram under construction.

With regards to data fields, the field Assignment date leads to adding an attribute named assignment_date to the class CLIENTORDER. Figure 18 depicts this derivation and Table 8 specifies the details of the new attribute. All attributes that are added to a class as a result of a class extension have the following properties: these attributes are not part of the identification function, the attribute type is Variable, they are not requested upon creation, and they allow nulls. The data type is derived from the domain of the field (in this case, the data type of assignment_date is Date because the field domain of Assignment date is date).

Table 8. Specification of the new attribute added to class CLIENTORDER

Attribute name	Id	Attribute type	Data type	Size	Requested	Null allowed
assignment_date	no	Variable	Date		no	yes

With regards to reference fields, the field Supplier references a business object that was processed in the communicative event SUPP 2. Therefore, a structural relationship is defined between the class CLIENTORDER and the class SUPPLIER. The cardinality is defined as 0:1 in the side of SUPPLIER because the orders are not assigned to suppliers when they are placed, but it occurs in a later moment in time. For the same reason, it is dynamic.

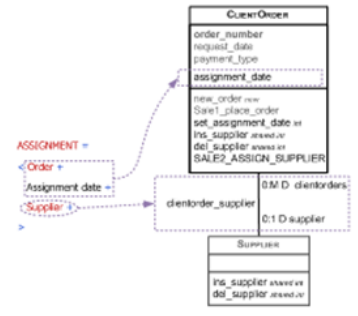
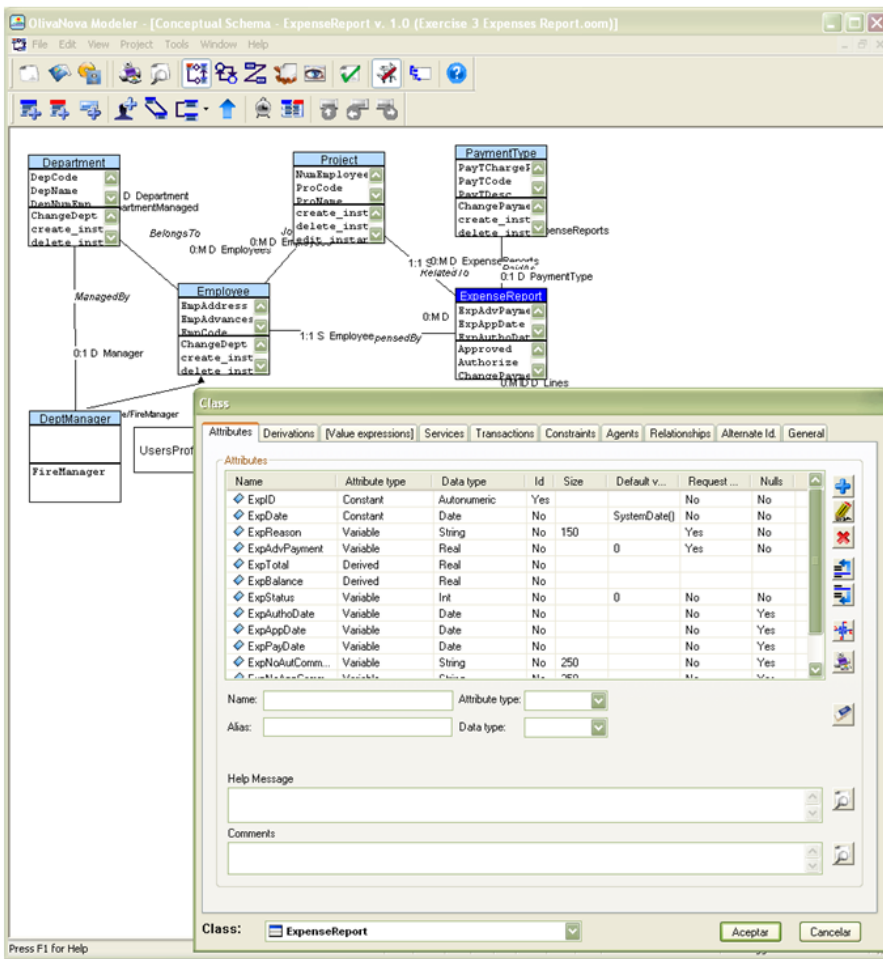


Figure 18. Class diagram view of SALE 2


Again, we have prepared both slides for oral presentation and textual material to be provided as a hand-out of the tutorial.

²² As argued above, as long as the communicative event diagram has been properly extended so as to include all the precedent communicative events and the requirements model is complete, the classes to be related already exist in the class diagram. In case of incompleteness, the classes may not exist.



The focus of the tutorial is put on methodological guidance, which is more applicable in industrial settings and across research lines, and not on notation, nor modeling tools, which are more dependent upon project contingencies and specific research agendas.

However, the tutorial will train in modeling tools, as a practical application of the methods and techniques.



OLIVANOVA Course Exercise. Expenses Report Management System

OLIVANOVA Course Exercise. Expenses Report Management System

1 Introduction

The Expense Report Management System is a tool that helps the process of expense reports to be followed. Using this tool, it is easy to follow the Expense Reports life cycle, from its creation until it is paid to the employee.

Employees give in their expense reports when all tickets relating to business travel or a business lunch have been collected.

If the employee received a payment in advance, this must be entered when the expense report is created. Besides, each expense will be specified in a line. The advanced payment must be subtracted from the total expense that will be paid to the employee.

When an Expense Report is issued, it must be reviewed by the Department Manager. Who will decide if the expense report is authorized and continues its flow. But he can deny the report, and specify the reason for not authorising the expense report.

Before paying an expense report, the payment must be approved by the Accounts Manager. He can also deny the payment, but in this case, he should indicate the reason why it is denied.

If the expense report is denied by a Department Manager or Accounts Manager, then the employee should modify it accordingly and depending on the reason given by the managers.

After the employee modifies the expense report, he should issue it again to start the process.

Finally, when an expense report is approved, an Accounts user or the Accounts Manager can pay it. Then, the expense report status will be "paid" and the payment date is kept.

2 Data definition of the system

2.1 Expense report

Header and footer: general information about expense. It collects all related expenses for a project and an employee.

- ✓ Report identifier. It must be unique.
- ✓ Creation date of the expense report.
- ✓ Employee that hands in the report
- ✓ Project related to these expenses
- ✓ Brief description of the expenses reason.
- ✓ Advance payment
- ✓ Total expenses: sum of all expenses
- ✓ Balance (Total expenses – Advance payment). If the balance is higher than 0, and the payment type contains an additional cost, this value will be subtracted. In case after subtracting the additional cost, the result of the operation is negative, then balance will be zero.

© CARE Technologies, S.A. 2009 1

Small exercises will be carried out during the tutorial, and additional material will be made available for further training.