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 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.

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.

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. OLIVANOVAS 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).
What does an information system do?

- An IS deploys an 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

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.
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.
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.

---

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 related 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, 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 data is kept.

2 Data definition of the system

2.1 Expense report

Headers and footers: 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.

---

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