Business process Eclipse Editor

Paolo Maresca, Armando Cotugno, Salvatore Mignogna
(University of Naples “Federico II”)

Roberto Longobardi, Alessandro Donatelli, Rosario Gangemi
(IBM Software Group Lab, Rome)
Software development: UCDD

- With **Use-Case Driven Development**, software requirements in the form of use-cases are captured in a use-case model from which subsequent software development artifacts are derived
  - Customers, users, developers (i.e. people coming from different fields) cooperate to define what software should do
- Traceability from use-cases down to software artifacts (and vice-versa) is granted
- Set of functionalities that are both desired (as specified by use-cases) and actually delivered is maximized
Potential UCDD issues

• How do we know that we have the right use-case model?
  – If we get the use-case model wrong the downstream artifacts will be incorrect as well and we will build the wrong system

• What happens to software requirements if something changes in the business layer?
  – We need a formal description of business specifications at the business layer
  – We need to trace business specification to software requirements in order to allow efficient reaction in application development to changes happening in the business layer

A solution: Process Driven Development
Process Driven Development (PDD)

“Functional requirements of individual users are largely determined by the business processes within which they operate”

- A clear vision of the business context that the software will support makes it possible to focus the real needs of customers and users
- Business processes within which the software will be used are formally modelled
- Software requirements are elicited straight from business process models
USBD: Unified Scenario-Based Design

- **Unified Scenario-Based Design (USBD)** is a Process Driven Development methodology developed at IBM Software Group Lab in Rome
- Business requirements are formally defined by means of business process models
- USBD:
  - brings software development process, thus developed software itself, to the real needs of business it will support
  - allows formal tracing between process models and system requirements
BEE project start-up

- Collaboration between IBM Software Group Lab in Rome and University of Naples “Federico II” to promote USBD

- Research:
  - “Comparing Business Definition Languages”
    - BPMN is the most widespread notation for business process modelling
  - “Comparative assessment of open source applications for business process management”
    - No open source applications try to fill the business-IT gap providing requirements elicitation facilities
  - Both researches refer to List-Korherr metamodel

- Development:
  - An application that supports USBD methodology
List-Korherr metamodel

- Collects concepts from both academic and industry fields
- Simple and complete
- Categorizes concepts in 5 perspectives:
  - Business Process Context
  - Behavioural
  - Functional
  - Informational
  - Organizational
Non sarebbe più opportuno parlare prima della visione di BEE e poi del metamodello di listkorherr in modo tale da non disperdere l'attenzione tra concetti dei quali parli, poi abbandonili per parlare di altro e poi riprendi in seguito? Inoltre non credi che sarebbe più opportuno approfondire di più il discorso descrittivo sul listkorher al fine di poter poi illustrare in modo più chiaro e completo le modifiche apportate allo stesso per giungere al metamodello di BEE? In merito alla behavioural perspective io suggerirei di descrivere a mezzo di cosa avviene la successione delle attività: "la behavioural perspective contiene i concetti inerenti il comportamento del processo aziendale in termini di successione delle attività per mezzo di flussi d’esecuzione e di elementi di controllo di tale flusso, oltre ai concetti inerenti il flusso informativo tra le diverse attività"
BEE vision

• Problem: develop an open source application that allows intuitive modelling of business processes and elicitation of software requirements
  – Easy to use by non-proficient users
  – Supports USBD, allowing requirements elicitation and traceability

• Solution: Business process Eclipse Editor
  – Specifically designed business process ontology (metamodel) based on List-Korherr metamodel
  – Intuitive notation based on BPMN
Metamodel development

- Concepts that are either unessential to BEE purpose or too complex for non-proficient users have been removed.
- Concepts that are typical of USBD methodology and needed for requirements elicitation and traceability features have been added.
- Intermediate result: Advanced List-Korherr metamodel.
- Next step: make metamodel suitable to model driven development.
BEE metamodel

- From 5 to 3 perspectives:
  - Business Process
  - Business Organization
  - Business Context
- Perspectives are aggregated in a Business Model
- BEE metamodel:
  - Is easy to read
  - Brings to a well-structured model editor
  - Well suits GMF editor generations
  - Is easy to extend
BEE development process

• Business process modelling for requirements elicitation is an innovative field
  – BEE requirements, metamodel and architecture needed to be adjusted during development

• BEE development team was heterogeneous and spread across different locations
  – Three IBM researchers from Rome as stakeholders, two students from Naples as analysts/developers, one professor from Naples as project leader

Agile Model-Driven Development process implemented using a collaborative development application
Leveraging IBM Rational Team Concert

- IBM Rational Team Concert (RTC) is a collaborative software development application based on IBM Jazz Platform
- RTC allowed us to coordinate project work, schedule and plan all project development steps, share and define rules and responsibilities
  - RTC Work-Items mechanism allowed us to control and manage project progress, monitor team members' work, assign tasks in a clear way and define tasks time constraints
  - RTC repository mechanism allowed us to easily share all project artifacts
BEE development with GMF

- Graphical Modeling Framework (GMF)
  - Framework for generating graphical editors based on EMF (Eclipse Modeling Framework) and GEF (Graphical Editing Framework)
  - Allows automatic generation of graphical editors based on a metamodel; each editor allows working either on the whole metamodel or on a specific portion of it
  - EMF component allows generation of (non graphical) model editors

- GMF used to generate BEE plug-ins
  - Automatic code generation from BEE metamodel
  - Graphical customization
  - Plug-ins code customization, constraints specification
BEE now: architecture and features

• Architecture
  BEE metamodel was used to develop, by mean of GMF, Java code realizing four Eclipse plug-ins that actually compose BEE feature:
  – Data model structure
  – Data model access features
  – Graphical editor
  – Non-graphical model editor

• Current release features
  – **Business Process Graphical Editor** allows modelling Business Process perspective concepts
  – Non-graphical **Business Model Editor** allows modelling all metamodel concepts
BEE in future: planned features

- Business process modelling
  - Several graphical editor each related to a specific metamodel perspective
  - BEE as multipage editor
- Requirements elicitation
  - Software requirements will be derived from process models
  - Requirements will be defined by mean of UML artifacts (Use-Case and Use-Case Realization)
  - UML artifacts produced in form of XMI files that can be later imported in UML modelling tool and CASE applications
- Traceability
  - By means of a traceability matrix that identifies correspondences between activities in the process model and use-cases
References

Beyond Use-Cases: Process Driven Development (PDD)
http://www.digerateur.com/visualocity/articles/processDrivenDevelopment.jsp

Unified Scenario-Based Version

An Evaluation of Conceptual Business Process Modelling Languages
http://wit.tuwien.ac.at/node/14042/publications/acmsac2006.pdf

Comparing business definition languages

Any questions?
http://www.nare.it/uni/beeproject/