A Foundation of Method Engineering and Self-Referential Enterprise Systems

Multi-Perspective Enterprise Modelling

Ulrich Frank

Information Systems and Enterprise Modelling

ICB Institute for Computer Science and Business Information Systems
<table>
<thead>
<tr>
<th></th>
<th>Enterprise Modelling: Foundational Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Method Engineering</td>
</tr>
<tr>
<td>3</td>
<td>Self-Referential Enterprise Systems</td>
</tr>
</tbody>
</table>
The Subject

- complexity
- social constructions
- technical artefacts
- beliefs & myths
- lack of understanding
- cultural chasm
- dissatisfactory practice
- change
- fear of change
Designing & Managing Business Information Systems: Some Objectives

- reduction of complexity, risk and costs
- fostering communication between management, users and IT-Experts
- promoting integration of
  - software systems
  - IT and business
- promoting reuse
- providing versatile tools for thinking
- fostering flexibility of the enterprise and its software systems
- coping with technological progress and horrors of the past
- creating sense of the enterprise and its IS
An **enterprise model** integrates at least one conceptual model of the information system (e.g. a class diagram) with at least one model of the relevant action system (e.g. a business process model).

A **multi-perspective enterprise model** is an enterprise model that emphasizes accounting for *perspectives*, which will usually correspond to professional views. These perspectives are represented in models constructed with domain-specific modelling languages (DSML).
Multi-Perspective Enterprise Modelling (MEMO)

- comprehensive approach to enterprise modelling
- development started in the early 1990s
- various components:
  - adaptable high-level framework
  - extensible set of DSML
  - reference models
  - language architecture
- supplemented by (meta) modelling environment (*MEMO Center*)

Generic Framework: „Map of the Enterprise“
Example: Multi-Language Diagram

Strategy Net

Business Process Map

IT Resource Diagram

Value Chain Diagram

Organisational Chart

Business Process Diagram

Object Model
DSML vs. GPML (2)

**DSML**

- **Server**
  - name: String
  - os: String
  - version: String

  Runs on
  - runs on

- **ERP**
  - name: String
  - version: String

  Runs on

  - runs on

**GPML**

- **Class**
  - name: String
  - isAbstract: Boolean

  Runs on

  - runs on

- **Attribute**
  - name: String

  Runs on

  - runs on

- **DBMS-Server**
  - os: String
  - version: String

  Runs on

- **SAP-R3**
  - os: String
  - version: String

  Runs on

- **M1**
- **M2**

© Ulrich Frank - Enterprise Modelling
Example: Decomposition of Value Chain

- Inbound Logistics
- Operations
- Outbound Logistics
- Marketing & Sales
- Service

- Contracting
- Compensation
- Risk Assessment

- Compensation Liability
- Compensation Cars
- Compensation Fire

- Claim not covered
- Prepare and send denial
- Claims denied

- Customer Attendant
- Check contract
- Claim covered
- Record details
Example: Goal Model

- **Maximize customer satisfaction**
  - **Satisfy sustainable sales goal in region A**
    - **Sustainable sales**
    - **Mar. 2012 2012-03-31**
    - **€5M**

- **Increase number of sales agents to 19**
  - **Sustainable sales**
  - **Mar. 2012 2012-03-31**
  - **€7M**

- **Keep labor costs at €8M**
  - **Labor costs**
  - **Non-std. 2012-02-15**
  - **€7M €5M**
  - **Mar. 2012 2012-03-31**

- **Satisfy sustainable sales goal in region B**
  - **Sustainable sales**
  - **Mar. 2012 2012-03-31**
  - **€12M**
DSML vs. GPML

© Ulrich Frank
Enterprise Modelling

Average Costs per month: 50,000 €
Average IT Costs per month: 15,000 €
Average IT Costs per instance*: 15 €

Processes
- Order Management
- Compensation
- Customer Acquisition
- Procurement
- Complaint Handling

Customer Rating
- Customer Data
- BI / DW

Customer Contact
- External Contact
- Contact Form

IT Organisation
- Oracle 10g
- DB Admin
- Application Server
- Database
- Server 4
- Backup Server
- Exchange
- Webserver
- IT Center Austin
- IT Center Munich, Germany
- Data Center Houston
- Data Center Georgia

Questions
- Is there any chance for outsourcing services?
Specific Challenges of DSML Design

Currently major re-launch of language architecture
The Essential Conflict of Designing DSML

Potential Productivity Gain
Scale of Reuse

Level of (domain-specific) Semantics
Method Engineering

- rationalist assumption: method approach of choice for addressing complex problems
- given set of methods not sufficient
- however: creating and validating methods require substantial expertise and effort
- hence, need for guiding the construction of methods

![What is a method?](linguistic structure of domain + process)

- Technical terminology
- DSML
Enterprise models enable elaborate, purposeful representations of problem domains in enterprises.

- on the level of reference models
- ... or on the level of DSML

Supplementing them with abstractions of processes/projects provides conceptual foundation for modelling (engineering) methods.

Existing modelling tools can be extended to support project monitoring and support.

Need for abstractions that cover a (wide) range of problem classes
Metamodel

Project
- name: String
description: String
similarInstances: Integer
lessonsLearnt: String
coreCompetence: Boolean
riskLevel: Level
strategicRel: Level
budget: Money

Phase
- name: String
description: String
risk: Boolean
intendedResults: String
defaultStart: Date
defaultFinish: Date
started: Date
finished: Date
result: String
resultAssessment: String

Role
- name: String
responsibility: String

Position
- name: String
averageSpan: Float
staff: Boolean

Resource
- name: String
description: String
unit: ResUnit
competitiveness: Level
qualityVariance: Level
cost: Level

Goal
- name: String
description: String
priority: Priority

Employee
- lastName: String
firstName: String
dateOfBirth: Date

ResAllocation
- reqQuality: Level
reqAvailability: Level
risk: Level
reqVolume: Float

Action
- name: String
description: String
successFactor: String

Method
- name: String
description: String
includes: Method

Diagram
- name: String
description: String
refersTo: Diagram

Diagram_State
- description: String
approved: Boolean

View on
- name: String
description: String
experience: String

© Ulrich Frank - Enterprise Modelling
This phase is aimed at determining requirements for enterprise software that result from intended support for business processes.

Assign abstract functions (independent of any implementation) to subprocesses within previously designed business process diagrams. Use function template. To infer functions perform ...

<table>
<thead>
<tr>
<th>Diagram</th>
<th>State</th>
<th>DSML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Process Diagram</td>
<td>revised</td>
<td>MEMO OrgML</td>
</tr>
<tr>
<td>Business Process Function Diagram</td>
<td>revised</td>
<td>MEMO OrgML, MEMO ITML</td>
</tr>
</tbody>
</table>
Example Execution of Method (Project)

- Clarify Strategy
- Model selected Business Processes
- Describe required Functions
- Model alternative Systems
- Perform Comparison
- Review and Refine Results
- Concluding Recommendations

Start: 13-02-04, Finish: 13-02-20
Start: 13-02-22, Finish: 13-03-04
Start: 13-03-04
Self-Referential Enterprise Systems: Motivation

- currently, (enterprise) modelling widely restricted to analysis and design phase

- use of enterprise software widely restricted to retrieving, manipulating and aggregating instance-level (M0) data
  - difficult to understand the system for users
  - almost impossible to modify it

- Enterprise software lacks representation of relevant context.

Integrating enterprise modelling environments with enterprise software seems promising.
Conception of Self-Referential Enterprise System

- integrates traditional enterprise systems (e.g. ERP systems) with corresponding enterprise model
  - enterprise model used at run time
  - supplements application system not only with conceptual model of itself, but also with model of its surroundings and purpose (goal system)
  - enrich models with access to components/data of corresponding enterprise system

- enables self-management and monitoring

- provides users with multiple navigation/analysis capabilities on different levels of abstraction

- enables advanced users to adapt a system by modifying models
Is there a chance for outsourcing services?
Example Navigation
Challenge: Mismatch of Abstraction Levels

Implied by limitations of prevalent programming languages.
... even worse

Synchronization of model and code hardly possible!
Characteristics of an Ideal Solution

- **common representation** of model and code
  - no need for code generation
  - Modifications become immediately effective.

- arbitrary number of classification levels possible

- beyond strict separation of classification levels as proposed by MOF
  - concepts on different classification levels in one model
  - implementation of deferred instantiation („intrinsic features“, „clabjects“, „powertypes“ etc.)

Prevalent programming languages no option.
Smalltalk as a Solution?

unfortunately not suited
XMF: eXtensible Metamodelling Facility

- meta programming language
- based on „golden braid“ architecture
- allows for arbitrary number of classification levels
- all (meta) classes are objects
- Classes have access to their instances at run-time – et vice versa.
- supplemented by (meta) modelling environment (Xmodeler)


satisfies all requirements.
Illustration of “Golden Braid” Architecture

- **Attribute**
  - ... 0,*
  - assigned to > specialized from > ... 0,*

- **Class**
  - name: String ... 0,*
  - new() ... 0,*
  - instance of > 0,*
  - 1,1
new MEMO language architecture based on “golden braid” model

extends XMF meta model

allows for representing all levels of abstraction/classification in one system

various update policies available, e.g.
- modification of class results in modification of all its instances
- modification of class results in modification of future instances only
- access to “deleted” features permitted or not
Current State and Perspectives

- prototypical implementation of SRES successful
- focus on business process models and respective process instances
- also: support for multi-level modelling
- creation and integration of new languages effectively supported by tools for generating editors from metamodels and specifications of concrete syntax
- XMF requires highly qualified programmers.
- governance recommended to avoid pitfalls
- joint project with Tony Clark to further develop the Xmodeler