Conceptual Modeling of Internet Sites
Part 2.5.: Engineering of WIS

Conceptual Modeling of Internet Sites
Wintersemester 2014, Gastprofessor@FH Hagenberg, Österreich

16.10.2014

Prof. Dr. Bernhard Thalheim
Technologie der Informationssysteme
Institut für Informatik, Christian-Albrechts-Universität zu Kiel, BRD
Kolmogorow-Professor e.h. der Lomonossow-Universität Moskau
The Software Engineering Triptych

- description of the application domain
- prescription of systems requirements
- specification of software systems

The WIS Engineering Scope

- behaviour of the user in the application domain
- behaviour of the human-computer interface
- behaviour of the software system
Implementation Choices: Architecture

Architecture-driven: Seeheim or Arch architecture

- Seeheim or Arch architecture (3-tier architecture)
  service-oriented layering with mediators

- Quasar architectures: application architecture, systems architecture, infrastructure architecture

- Monolithic architectures

Mediators Between Users, HCI, IS

Architecture-driven: Seeheim or Arch architecture (3-tier architecture)

- user-HCI: presentation media

- HCI-IS: media types, containers, stories
The Software Engineering Triptych

Description/prescription layer

Conceptual layer

Implementation layer

Application domain description

Requirements prescriptions

WIS utilisation

Presentation system specification

Information systems specification

WIS specification

Presentation system

Information system

Web information system
The Principal Element Types of Co-Design and SPICE

**CO-DESIGN**
- Aspect
  - Layer
    - Step
      - Document

**SPICE Process Dimension**
- Process Group
  - Process
    - Base Practice
    - Output Work Product
      - Input Product

**SPICE Capability Dimension**
- Capability Level
  - Process Attribute
    - Generic Practice
    - Generic Work Product
      - Generic Resource
Combining with Co-Design Abstraction Layers

Application domain layer

Scoping

Requirements acquisition layer

Variating

Business user layer

Designing

Conceptual layer

Implementing

Implementation layer

Structuring specification

Distribution specification

Dialogue specification

Functionality specification
Making Co-Design Working: General Description

Application domain or strategic layer:

HERM (concept map (concept)),
HERM (functionality (feature)),
DistrLang (contract sketch (contract, quality criteria))
SiteLang (application story (application step))

(1) Developing visions, aims and goals
(2) Analysis of challenges and competitors

Stakeholder contract ("Lastenheft")
nowadays: Product feature catalog
Making Co-Design Working

Requirements Specification

Requirements acquisition layer:

- HERM (sketch (rough type)),
- HERM (business process (business step)),
- DistrLang (contract opportunities),
- SiteLang (story (event))

(3) Separation into system components
(4) Sketching the story space
(5) Sketching the view suite
(6) Specifying business processes

IS development and system documentation (“Pflichtenheft”)

Making Co-Design Working

Usage, Usability and Application Stories

Business user layer:

- HERM (skeleton (application type)),
- HERM (activity (action)),
- DistrLang (media types, contract),
- SiteLang (plot (theme), actors, media types)

(7) Development of scenarios of the story space
(8) Elicitation of main data types and their associations
(9) Development of kernel integrity constraints, e.g., identification constraints
(10) Specification of user actions, usability requirements, and sketching media types
(11) Elicitation of ubiquity and security requirements

Playout system specification
Making Co-Design Working

Conceptual Specification

Conceptual layer:

- HERM (schema(types)),
- HERM (workflow (process)),
- DistrLang (service space, exchange frame),
- SiteLang (story space, actors, media types, presentation)

(12) Specification of the story space
(13) Development of data types, integrity constraint, their enforcement
(14) Specification of the view suite, services and exchange frames
(15) Development of workflows
(16) Control of results by sample data, sample processes, and sample scenarios
(17) Specification of the media type suite
(18) Modular refinement of types, views, operations, services, and scenes
(19) Normalization of structures
(20) Integration of components along architecture

Conceptual schemata
Making Co-Design Working

**Implementation Layer**

**Implementation** layer:
- o-r DDL (o-r schema (relation)),
- PL (module) language (program, trigger, sp, ...),
- Distribution specification language (distributed system (distribution, protocol, call))
- Dialog system language (presentation space (working sheet))

(21) Transformation of conceptual schemata into logical schemata, programs, and interfaces
(22) Development of logical services and exchange frames
(23) Developing solutions for performance improvement, tuning
(24) Transformation of logical schemata into physical schemata
(25) Checking durability, robustness, scalability, and extensibility

Program library
complexity ⇔ common misbelief in existence of a theory of CM
The Work Product Development Dimension

The main product is the model; it shall however used for realisation!

The work product

depends on abstraction layer e.g. requirements, specification, realisation or implementation layer,

depends on granularity and precision of the work product itself,

depends on resources used for development such as the languages,

depends on level of separation of concern such as static/dynamic properties, local/global scope, facets,

depends on quality properties of the input, e.g. requirements, completeness, conciseness, coherence, understandability,

depends on decomposition of the work products in ensembles of sub-products, and

satisfies quality characteristics

quality in use

internal quality
The Knowledge Gap for the Classical Information Systems Development Process

“Partial reality”

- Part of reality
- Things of reality
- Observed property
- “Topic”
- Predicator

Foundation of decisions - Modelling decision - Context

Usage of theory - Revision during the development process - acts within

Modality - “Schema” as result and partial point of view of a database development process

Exactness - Confidence
Typical life scenario

- Somebody asks for directions
  same language, same local landmarks, full information (any small street), request understanding

- Somebody searches for a hotel
  property-based search
  association-based search
  zapping, browsing, offer hunting

- Somebody is going to be relocated
  complex bundle of content and functionality depending on the communities, the customers, the life circumstances
Integration der Beschreibung des Anwendungsgebietes

Lebens- und Anwendungsfälle, Fachverfahren
Application Domain Description

(1) modelling business processing facets of an application domain,

(2) modelling intrinsic facets of an application domain,

(3) modelling possible support technology facets of an application domain,

(4) modelling possible management and organisation facets of an application domain,

(5) modelling possible rules and regulations of an application domain,

(6) modelling possible script (story) facets of an application domain,

(7) modelling possible human behavior facets of an application domain
Application Domain Description (1)

Users → Intentions → Life Cases → Context

Profiles

Brand of WIS (scope)

Portfolio

Word fields → Associates

Substantive word fields

Verb word fields

Media types

Containers → Presentation

Storyboard

Application Domain Description

Containers

Concept

Topic

Information

Content

Overview

Triptych

Life cases

Profiles

Portfolio

Context

Requirements
**Application Domain Description (2)**

- D. Bjorner: High-level, informal application domain description
- L. Heinrich: Strategic information analysis
- Volere: Business use cases (simple word fields) combined to business scenarios and business tasks
Modelling Life Cases by Stories

Digicult: Representing successful behavioural pattern

Analogous search
- Deleting options
- No further options
- Gaining info on problems
- Survey on opportunities
- Sample cases
- Similar cases
- Successful cases
- Approaches for use
- Background info

Mapping behaviour of users with full option space
Intelligent representation of information and knowledge spaces
Adaptation to the user, current situation, context, ...
Logistics for content
User Intentions

Intention space: ⟨intention name⟩
Purpose: ⟨outcome description⟩
Aims: ⟨list of aims⟩
Objectives: ⟨list of objectives⟩
Intents: ⟨outcome description⟩
Targets: ⟨list of weighted targets⟩
Objects: ⟨list of weighted objects⟩
Themes: ⟨class of intents⟩
Time: ⟨outcome description⟩
Design: ⟨general flow⟩
End: ⟨effects, termination conditions⟩
Occasion: ⟨list of objectives⟩
Representation: ⟨general style guide⟩
Atmosphere: ⟨general description of atmosphere⟩
Metaphors: ⟨list of metaphors⟩
Based On: ⟨tasks, audience, mission, goal⟩
Input for the Application Domain Description

- User: General description of the users
- User intentions: General description gathering the reasons to visit the WIS
- User profiles: General specification of the users together with personas
- Context of the WIS in general
Application Domain Description

- Brand of the WIS
- Life cases with relative importance for the WIS
- Word fields as basis business activities (business use cases and events)
- Associators representing the general life case chart
- Portfolio supported by the WIS
- Context of the WIS depending on technology
- Actor profiles and portfolio
Life Cases

Life case: ⟨life case name⟩
Characterisation: ⟨outcome description⟩
  Tasks: ⟨list of user tasks⟩
  Problems: ⟨list of problems⟩
  Background: ⟨general characterisation⟩
  Objectives: ⟨list of objectives⟩
Life case flow: ⟨general graphical description⟩
  Milestones: ⟨graph of milestones⟩
  Content consumed: ⟨consumed content items⟩
  Content produced: ⟨produced content items⟩
  Themes: ⟨class of intents⟩
Figures: ⟨actors list⟩
  Expected profile: ⟨general profile description⟩
  Collaboration: ⟨general collaboration description⟩
Context: ⟨general context description⟩
  Time: ⟨temporality limitations⟩
  Place: ⟨assignment of places⟩
  Legacy: ⟨names of documents⟩
  WIS: ⟨general WIS context⟩
Representation: ⟨general behavior⟩
  Approaches: ⟨general description of approaches⟩
Refinement of Life Cases

Life case: \langle \text{life case name} \rangle

Tasks: \langle \text{list of tasks associated with the life case} \rangle

Actors involvement: \langle \text{description of actors involvement} \rangle

Profile restrictions: \langle \text{list of restrictions due to actors profile} \rangle

Context specialisation: \langle \text{context embedding} \rangle
Life Case Elicitation

- Abstraction by making useful generalisations
- Business events investigation observing business reality
- Brainstorming for generating good ideas and to solve problems
- Community therapy by bounding stakeholders together like a family steps: intake for scoping, meaning elicitation, significance, response
- Interviewing for gathering
- Mind mapping by taking extensive and meaningful notes
- Neurolinguistic programming based on models, skills and techniques for thinking and acting effectively
- ...

...
Life Case Elicitation

- ...
- Reusing designs, requirements, and ideas
- Simulation model scenarios and prototyping based on life stories
- Soft systems observation and modelling for understanding and tackling real-world problems
- Systems archeology for extracting ideas, requirements and design from existing systems
- Use case workshops by reviewing and questioning based on business use cases
- Video recording of user reactions to software products
- Viewpoints focusing...
Profiles

User profile: ⟨user profile name⟩
Education profile: ⟨general description⟩
  Education:
  Capabilities:
  Knowledge:  ⟨description of knowledge in the application⟩
Work profile: ⟨general description⟩
  Task expertise:  ⟨description of knowledge⟩
  Task experience:  ⟨positive and negative experience⟩
  System experience:  ⟨experience with infrastructure planned⟩
Information profile: ⟨information need⟩
Interaction profile: ⟨interaction properties⟩
Personality profile: ⟨general description⟩
  General properties:  ⟨list of user properties⟩
  Preferences:  ⟨list of input/output/dialogue preferences⟩
  Polarity profile:  ⟨list of polarity properties⟩
Derivable profiles: ⟨profile description and enforcement⟩
  Security profile:  ⟨access control and privacy⟩
  Safety profile:  ⟨safety requirements⟩
Based On: ⟨user goals⟩
Based On: ⟨user types⟩
Actor Profiles

Actor profile: ⟨actor profile name⟩
Grouping criteria: ⟨characteristics of grouping of users⟩
Information demand: ⟨general description⟩
Utilisation pattern: ⟨general description⟩
Specific utilisation: ⟨general description⟩
Actor context: ⟨general description⟩
Portfolio

Actor portfolio: \langle actor portfolio name \rangle
Task: \langle general description \rangle
  Extension of: General characterisation of tasks
  Characterisation: \langle general description \rangle
    Initial state: \langle characterisation of the initial state \rangle
    Target state: \langle characterisation of the target state \rangle
    Profile: \langle profile presupposed for solution \rangle
    Instruments: \langle list of instruments for solution \rangle
    Collaboration: \langle specification of collaboration required \rangle
    Auxiliary: \langle list of auxiliary conditions \rangle
  Execution: \langle list of activities, control, data \rangle
  Result: \langle final state, satisfied target conditions \rangle
  Actors involvement: \langle general description \rangle
    Role: \langle description of role \rangle
    Part: \langle behavioural categories and stereotypes \rangle
  Collaboration: \langle general description \rangle
    Communication: \langle protocols and exchange \rangle
    Coordination: \langle contracts and enforcement \rangle
    Cooperation: \langle flow of work \rangle
  Restrictions: \langle general description \rangle
    Actor restrictions: \langle general description \rangle
    Environment: \langle general description \rangle
  Based On: \langle life cases \rangle
  Based On: \langle intentions \rangle
  Based On: \langle general tasks, audience, mission, goal \rangle
Context

Context:
  Extension of: (context name)
  General context 
Actor context:
  Projection context: (general description)
  Approximation context: (expectations)
  Ambiguity context: (condensations and abstractions)
  Mental state context: (scope)
  Characterisation: (general description)
Storyboard context:
  Pre-scene context: (general description)
  Post-scene context: (history of usage)
  Scene context: (potential continuation)
WIS context:
  Source and acquisition: (superimposed meta-data)
  Associated content: (system environment)
  Supported functionality: (content environment)
  Security: (function environment)
Temporal context:
  Versioning: (required security functionality)
  Development phase: (general description)
Provider context:
  (general description)
Developer context:
  (general description)
Organisational and social context:
  (general description)
Based On:
  Based On: (life cases, portfolio)
  Based On: (scenarios)
  Based On: (general tasks, audience)
Input of the Application Domain Description to Requirements Prescription

- Brand of the WIS
- Life cases with relative importance for the WIS
- Word fields as basis business activities (business use cases and events)
- Associators representing the general life case chart
- Portfolio supported by the WIS
- Context of the WIS depending on technology
- Actor profiles and portfolio
Requirements Prescription

- Content chunks representing content, concepts and topics
- Function chunks representing all functionality requirements
- Storyboard of the WIS
- Media type requirements (data, content, functions, containers, performance, operational, maintainability, support, security)
- Representation requirements (look and feel, grids and pattern)
- Storyboard requirements (usability, humanity, privacy, cultural, political, legal)
Mappings

- Decomposing and synthesising life cases to word fields and associators of word fields
- Associators or rough stories to storyboards
- Word fields to chunks (data and/or functions)
- Chunks to media types
- Chunks and storyboards to containers
- Storyboard to design patterns and grids
References

- L. Heinrich, Informationsmanagement. Oldenbourg, 1997
- www.systemsguild.com
- Schewe/Thalheim and co-authors
  
  see next slides
**Publications on Co-Design**


- Schewe, K.-D.; Thalheim, B.: Towards a theory of consistency enforcement. Acta Informatica, 36, 1999, 97-141. *Instead of falling into the traps of rule triggering systems*


Publications on Web IS Engineering


Publications on Science and Art of Conceptual Modelling


Publications on Model Suites, Evolution, Migration


Publications on Pattern Development


Publications on Component Development


Publications on Content Management


Publications on Privacy


