Erfahrungen mit dem Experience-Factory-Ansatz

Klaus-Dieter Althoff & Jens Mänz

Intelligent Information Systems
University of Hildesheim
Email: althoff|maenz@iis.uni-hildesheim.de
Überblick

• Experience Factory (EF)
• Case-Based Reasoning (CBR)
• Integration von EF und CBR (DISER-Methode)
  – Ergebnisse (Dissertation Carsten Tautz 2001)
• Weiterentwicklung (Erfahrungsbasierte Informationssysteme; EbIS)
  – Ergebnisse (Dissertation Markus Nick 2005)
• Ausblick
  – Arbeiten an der Universität Hildesheim
    • teilweise in Kooperation mit dem Fraunhofer IESE
Experience Factory (EF)
Case-Based Reasoning (CBR)

An approach to solve new problems by adapting solutions of similar past problems.

Problem: Initial problem description defines new case
Retrieve: New case is used to find a similar case in the case base
Reuse: Combination of new and retrieved case provides solved case
Revise: Evaluation of the suggested solution
Retain: Learning of useful experience through adapting the case base and/or the general domain knowledge

*nach Aamodt/Plaza (1994)*
Relating CBR and EF/QIP
Evaluation – Experimental Results*

• EbIS approach (»Using the EbIS«) versus human-based approach (»Talking to colleagues«)
• The experiment showed:
  – Efficiency:
    • The EbIS approach finds more useful guidelines and observations per time period (in terms of both effort and duration).
  – Effectivity:
    • The EbIS approach finds useful guidelines and observations not obtained by the human-based approach.
• The experiment validated this in a statistically significant way.
  – Result:
    • **Combine human-based and EbIS approach**
• The participants agreed: 28 out of 29 participants would apply both approaches in combination.

EbIS-Product-Line Architecture

- **User Interface**
  - Data Entry & Data Editing
  - Reports
  - Intelligent Technologies
  - Maintenance Interface

- **Application Logic**
  - Intelligent Technologies
  - EbIS Maintenance Support

- **Database Management System**

- **Data Storage**

Special EbIS tools

“Traditional” tools/systems/infrastructure
Components of EbIS Product-Line

- **Intelligent Search**
  - RAISIN/1 (IESE)
  - RAISIN/3gta (IESE)
  - Orenge (COTS from empolis)

- **Browsing**
  - TaxBrow – Taxonomy Browser (IESE)
  - ModelExplainer (IESE)

- **Aggregation**
  - Context Aggregator (IESE)

- **Data Entry, Editing, and Reports**
  - J2EE-based technology
  - Microsoft Access (cheap COTS)
  - Eclipse IDE plugins
  - Microsoft dotNET IDE plugins [planned]

- **Server/Container**
  - Apache Tomcat as J2EE container (open source)
  - IBM’s Eclipse as Integrated Software Development Environment (open source, Java)
  - *Microsoft Access*
  - Microsoft dotNET [planned]

- **Database Management System**
  - Microsoft Access (cheap COTS)
  - MySQL (open source)
  - PostgreSQL (open source)
  - Microsoft SQL Server
  - Oracle (high-end solution)

- **Maintenance Support**
  - EMSIG tools for evaluation and maintenance
EbIS Development Process

EbIS Development with DILLEBIS

- DISER
  - requirements
  - environmental factors
- reuse of application-independent concepts
- Rapid EbIS Design
- pattern-based EbIS concept
- EbIS Patterns
- DISER
- maintenance knowledge acquisition
- tailored EbIS concept
- tailoring of evaluation program
- implementation
- test
- deployment
- deployed & running EbIS

INTERESTS
- reference architecture
- schema guidelines
- components

learn
Evaluation - Applicability within Projects

- **Broad applicability of EbIS method/tool**: Occurrence in successful EbISs in real-world projects
  - EbISs of different size and project type (→ “breadth”)
  - EbIS successful
    - Status “accepted“ (usage >= 1 year) OR
    - Status “deployable“ and EbIS tightly integrated
      - **Assumption** for tight integration: Acceptance and correct usage of the tool that supports the business process

<table>
<thead>
<tr>
<th>Status</th>
<th>#</th>
<th>Size of EbIS (#User, #Cases)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Small</td>
</tr>
<tr>
<td>Successful</td>
<td>8</td>
<td>SKe-Pilot, ISI, SLI-EB, IPQM, KM-PEB, CBR-PEB</td>
</tr>
<tr>
<td>Implementation</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Status unknown</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>#successful</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>
Nutzen

• Prototypische Realisierung von EF/EB mit CBR-Technologie
• Entwicklung einer systematischen, umfassenden Methode zur Entwicklung von EF/EB (DISER)
• „Technologieabstraktion“: EbIS
  – Einbeziehung alternativer und weitergehender Technologien
• Organisatorische Einbettung von CBR-Systemen in Industrie und Verwaltung
• (Zielorientierte) Evaluation und Wartung von CBR-Systemen
Ausblick

• Nutzung detailliert beschriebener Methoden (DISER/DILLEBIS) zur (teilweisen) Automatisierung solcher Prozesse
  – Beispiel: Intelligente Informationssysteme für Anwendungen im Ambient-Intelligence-Bereich
• Ziel: Integration von EF/CBR mit Software-Produktlinien (SPL)
  – Fokus auf „Wissen“
  – Nutzung von Agententechnologie zur Modularisierung auf der Wissensebene
  – Realisierung von einzelnen Agenten als EF/CBR-System
  – SPL als (hierarchische) EF solcher EF/CBR-Systeme
  – Beispiel: Simulation von Unternehmensgründungsprozessen mit Multiagentensystemen
Softwareagentenbasierte EF

Projekt A
Projekt B
Projekt C

SE-Beratungsorganisation
SE-Portal
Fallbasis

Evaluationsagent
Erfahrungsingenieuragent
Wartungsagent

SE-Berater-Agent

learn
reuse
Benefit - Case Study with Students*

- **Ziel:** Evaluation der initialen Konzepte der EbIS-Entwicklungsmethode
- Zwei Rollenspiele mit jeweils sieben Studenten mit nahezu identischem Design
  - weitere Rollenspiele mit ähnlichem Design in anderen Semestern
- **Rollenspiel:**
  - **Kontext:** Organisation zur Entwicklung von CBR-Anwendungen (X-CBR)
    - Situation und Historie zu X-CBR
  - **Aufgabe:**
    - Das X-CBR-Management entscheidet, dass eine Experience Factory über CBR-Projekte aufgebaut werden soll, um das Wissen zur Kernkompetenz „CBR-Anwendungen“ besser managen zu können
    - Übernahme der EF- und Organisationsrollen durch Dozent, Mitarbeiter und Studenten
    - Initielle Modellierung des EbIS
    - Ausfüllen eines abschließenden Fragebogens
    - Feedback-Runde
- **Ergebnisse:**
  - Initielle Modellierung in 180 min (2 Vorlesungsblöcke á 90 min)
  - Praktische Erfahrung für initialen Workshop mit Industriepartnern (Finanzdienstleister)
  - Einführung des Phasenkonzeptes in der EbIS-Entwicklungsmethode

Erfahrungen mit dem Experience-Factory-Ansatz

Klaus-Dieter Althoff & Jens Mänz

Intelligent Information Systems
University of Hildesheim
Email: althoff|maenz@iis.uni-hildesheim.de
Overview

- Case-Based Reasoning (CBR)
- Experience Factory (EF)
- Integrating CBR and EF:
  - DISER method
  - Evaluation of DISER
  - Improvement of DISER: DILLEBIS method
  - Evaluation of DILLEBIS
- Benefits
  - from an SE perspective
  - from an AI perspective
- Implications and Outlook
Case-Based Reasoning (CBR)

An approach to solve new problems by adapting solutions of similar past problems.

Problem: Initial problem description defines new case
Retrieve: New case is used to find a similar case in the case base
Reuse: Combination of new and retrieved case provides solved case
Revise: Evaluation of the suggested solution
Retain: Learning of useful experience through adapting the case base and/or the general domain knowledge

* nach Aamodt/Plaza (1994)
problem solving and learning from experience

retrieve

reuse

revise

retain

case-based reasoning

collect descriptors

identify features

search

initially match

select

adapt

evaluate solution

repair fault

extract index

extract relevant descriptors

extract solutions

extract solutions

extract justification

evaluate in real world

evaluate in model

self-repair

user-repair

adjust indexes

develop indexes

determine indexes

update general knowledge

rerun problem

generalize indexes

determine indexes

update general knowledge

evaluate by teacher

repair fault

infer descriptors

interpret problem

infer descriptors

evaluate in real world

evaluate in model

self-repair

user-repair

adjust indexes

develop indexes

determine indexes

update general knowledge

evaluate by teacher

repair fault
Abstracted Method for the Retain Task

IF no_similar_past_case (current_case)
    THEN construct_new_case;
    ELSE lazy_generalise (old_case);
IF current_case_successful
    THEN integrate_into_successful_cases;
    ELSE integrate_into_total_problem_cases;
DO adaptation UNTIL system_behaves_as_wanted
Case Ontology for IESE Experience Factory

Diagram:
- Artifact
- Process
- Improvement Suggestion
- Project
- Problem
- Pragmatic Solution
- Observation
- Guideline
- Lesson Learned

Relationships:
- defines
- identifies
- is-a
- has-part
- has-decomposition
Experience Factory (EF) and Quality Improvement Paradigm (QIP)

Quality Improvement Paradigm
(Basili, Rombach, 1988)

Experience Factory Organization
(Basili, Rombach, 1988)
Experience Factory (EF)

**Project Organization**

**Plan**
1. Characterize
2. Set Goals
3. Choose Model

**Perform**
4. Execute Project

**Learn**
5. Analysis
6. Packaging

**Experience Engineer**

**Knowledge from past projects**

**Cases General Knowledge Feedback**

**Experience Base**

**Experience Factory**

**Project Team**

**Project N**

**Project 1**
Experience Factory Roles

Experience Factory Roles

Consolidated description of eight different EF roles

Experience Factory Roles

Experience Management

Knowledge Infusion

Project Support

Experience Factory Manager

Experience Engineer

Experiment

Librarian

Technology Expert

Candidate

Experience Base

Project Data Base

Experience Engineer

Experience Management
goals
design
reports
assets
experience packages
reusable experience
a project
data & lessons learned

experience
packages

Technology Expert

Candidate

Experience Base

Project Data Base
Relating CBR and EF/QIP
DISER Method
EbIS Aufgaben & Methoden: Conceptualize

- define concepts
- define nonterminal concept attributes
- identify reuse information
- define global similarity
- define dependencies
- define maintenance policies

- determine reusability factors
- classify reusability factors
- determine minimal quality requirements
- determine application boundaries
- define application policies
- acquire distinguishing characteristic concepts
- define terminal concept attributes

- define meaning of similarity
- identify reusability factors

- classify attribute
- define type

- define value range
- define local similarity
Conceptualize (1)

- **Goal:** Conceptualization of EbIS content
- **Input:** Reuse scenarios
- **Output:**
  - Schema of EbIS (conceptual model in the sense of structure based CBR)
  - Feedback indicators
  - Minimal quality requirements for reuse
  - Processes for reuse
Conceptualize (2)

• **Decomposition:**
  – Define concepts
  – Define nonterminal attributes
  – Identify reuse information
  – Define global similarity
  – Define dependencies
  – Define maintenance policies

• **Method:** Subtasks are carried out sequentially or iteratively.
Evaluation – Experimental Results*

- EbIS approach (»Using the EbIS«) versus human-based approach (»Talking to colleagues«)
- The experiment showed:
  - Efficiency:
    - The EbIS approach finds more useful guidelines and observations per time period (in terms of both effort and duration).
  - Effectivity:
    - The EbIS approach finds useful guidelines and observations not obtained by the human-based approach.
- The experiment validated this in a statistically significant way.
  - Result:
    - Combine human-based and EbIS approach
- The participants agreed: 28 out of 29 participants would apply both approaches in combination.

---

Deficiencies of DISER

- phase models and development strategies for a better integrability on the software process side;
- solutions for “feedback loops” as well as experience life cycle models;
- solutions for relating different types of knowledge/experience each represented on a different level of granularity;
- rapid application development approaches for a „cheap start“;
- knowledge modeling approaches/guidelines for “scaling up”;
- scalability of the underlying knowledge technology;
- integrability of knowledge technology with traditional software system technology;
- supporting maintenance as a knowledge-intensive task;
- maintenance process;
- decision support for maintenance;
- acquisition method for maintenance knowledge;
- maintenance enactment support (for optimizing the maintenance process);
- business goal oriented method for running an EbIS;
- relating maintenance to the goals of an EbIS to guide maintenance with evaluation;
- availability of an evaluation plan and maintenance knowledge already for the beginning of regular use for handling the to be expected continuous stream of experience.
EbIS-Product-Line Architecture

- User Interface
  - Data Entry & Data Editing
  - Reports
  - Intelligent Technologies
  - Maintenance Interface

- Application Logic
  - Intelligent Technologies

- Database Management System

- Data Storage

-EbIS Maintenance Support

Special EbIS tools

“Traditional” tools/systems/infrastructure
Components of EbIS Product-Line

- **Intelligent Search**
  - RAISIN/1 (IESE)
  - RAISIN/3gta (IESE)
  - Orenge (COTS from empolis)

- **Browsing**
  - TaxBrow – Taxonomy Browser (IESE)
  - ModelExplainer (IESE)

- **Aggregation**
  - Context Aggregator (IESE)

- **Data Entry, Editing, and Reports**
  - J2EE-based technology
  - Microsoft Access (cheap COTS)
  - Eclipse IDE plugins
  - Microsoft dotNET IDE plugins [planned]

- **Server/Container**
  - Apache Tomcat as J2EE container (open source)
  - IBM’s Eclipse as Integrated Software Development Environment (open source, Java)
  - *Microsoft Access*
  - Microsoft dotNET [planned]

- **Data Base Management System**
  - Microsoft Access (cheap COTS)
  - MySQL (open source)
  - PostgreSQL (open source)
  - Microsoft SQL Server
  - Oracle (high-end solution)

- **Maintenance Support**
  - EMSIG tools for evaluation and maintenance
Evaluation - Applicability within Projects

- **Broad applicability of EbIS method/tool**: Occurrence in successful EbISs in real-world projects
  - EbISs of different size and project type (→ “breadth”)
  - EbIS successful
    - Status “accepted“ (usage >= 1 year) OR
    - Status “deployable“ and EbIS tightly integrated
      - **Assumption** for tight integration: Acceptance and correct usage of the tool that supports the business process

<table>
<thead>
<tr>
<th>Status</th>
<th>#</th>
<th>Size of EbIS (#User, #Cases)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Small</strong></td>
</tr>
<tr>
<td>Successful</td>
<td>8</td>
<td>SKe-Pilot, ISI, SLI-EB, IPQM, KM-PEB, CBR-PEB</td>
</tr>
<tr>
<td>Implementation</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Status unknown</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>#successful</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Medium</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>indiGo/CoIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ESERNET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T-Com SR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project A</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Large</strong></td>
</tr>
</tbody>
</table>

33
Benefits

• Benefits from SE perspective
  – Prototyping a solution for EF
  – Developing DISER based on this prototype
  – Abstracting/generalizing from CBR systems to EbIS
  – ...

• Benefits from AI perspective
  – Evaluation approach for CBR/KBS
  – Real-life method: used and integrated into the work process
  – Need for additional AI techniques
  – ...
Implications and Outlook

• Integration of
  – Multi-agent systems
  – CBR
  – EF
  – Software product-lines

• Building intelligent information systems for supporting Ambient-Intelligence-like scenarios
Benefit - Case Study with Students*

- **Ziel:** Evaluation der initialen Konzepte der EbIS-Entwicklungsmethode
- Zwei Rollenspiele mit jeweils sieben Studenten mit nahezu identischem Design
  - weitere Rollenspiele mit ähnlichem Design in anderen Semestern
- **Rollenspiel:**
  - Kontext: Organisation zur Entwicklung von CBR-Anwendungen (X-CBR)
    - Situation und Historie zu X-CBR
  - **Aufgabe:**
    - Das X-CBR-Management entscheidet, dass eine Experience Factory über CBR-Projekte aufgebaut werden soll, um das Wissen zur Kernkompetenz „CBR-Anwendungen“ besser managen zu können
    - Übernahme der EF- und Organisationsrollen durch Dozent, Mitarbeiter und Studenten
    - Initiale Modellierung des EbIS
    - Ausfüllen eines abschließenden Fragebogens
    - Feedback-Runde
- **Ergebnisse:**
  - Initiale Modellierung in 180 min (2 Vorlesungsblöcke á 90 min)
  - Praktische Erfahrung für initialen Workshop mit Industriepartnern (Finanzdienstleister)
  - Einführung des Phasenkonzeptes in der EbIS-Entwicklungsmethode

EbIS Development Process

EbIS Development with DILLEBIS

DISER
- requirements
- environmental factors

Rapid EbIS Design
- reuse of application-independent concepts

pattern-based EbIS concept

EbIS Patterns

learn

DISER
- maintenance knowledge acquisition
- tailoring of evaluation program

tailored EbIS concept

INTERESTS
- reference architecture
- schema guidelines
- components

implementation
- test
- deployment

deployed & running EbIS
Running an EbIS with DILLEBIS

- development & deployment
- deployed & running EbIS
  - evaluation program
    - link for guidance
  - maintenance knowledge
- EMSIG process and tools
- corrections & improvements
  - evaluation (incl. monitoring)
    - guidance
    - maintenance
Research Areas in AI and SE and Their Intersections

Artificial Intelligence
- Learning
- KA
- DAI
- CBR
- Aml
- CI
- Agents
- KBS

Software Engineering
- DM
- PM
- AOSE
- RE
- DE
- CE

Fig. 2
EF Roles

• The **manager** provides resources, defines strategic goals and initiates improvement programs.
  – He determines the structure and content of the case base and controls its quality.
• The **supporter** is responsible for documenting new experiences and supporting the project team.
  – He collects and qualifies artifacts from the projects in accordance with the reuse criteria and the goals of the engineer.
  – Upon request, he supports the project team in retrieving and modifying the experience knowledge.
• The **engineer** is responsible for packaging and analyzing existing experiences.
  – Together with the manager, he identifies new reuse criteria and, based on that, acquires new cases.
  – He analyzes the case base in order to detect (further) improvement potential.
• The **librarian** is responsible for technical aspects like setting-up and maintaining the case base, storing, and publishing new cases.