AIFB

(c) Institut AIFB, Universität Karlsruhe
SS 2001

Kapitel 4

Case-based Reasoning (CBR)

Sources, partially used for preparing this presentation

- http://www.cs.auckland.ac.nz/~ian/teaching.html
  - Ian Watson
    Dept. of Computer Science
    University of Auckland

- http://wwwagr.informatik.uni-kl.de/sections/education/index.html
  - Dr. rer. nat. Ralph Bergmann
  - Dr. rer. nat. habil. Klaus-Dieter Althoff

    University of Kaiserslautern
    Department of Computer Science

- http://www.aic.nrl.navy.mil/~aha/slides/
  - David W. Aha
    Navy Center for Applied Research in AI
    Naval Research Laboratory

Thanks to authors!

AIFB

4.1 Case-based Reasoning
4.1.1 CBR - basis

- CBR is relative new discipline in AI and is commonly described as
  - a problem-solving method or reasoning model
  - whose core processes revolve around the
    - retrieval,
    - reuse, and
    - retention of previously encountered problem-solving episodes or cases.
- CBR technology has been applied to a variety of
  - analytic (e.g. classification, prognosis, decision support) and
  - synthetic (e.g., design, configuration, planning) problem-solving or decision-making tasks.

Definitions

- A case- based reasoner solves new problems by adapting solutions that were used to solve old problems (Reisbeck & Schank 1986)
- CBR is both the ways people use cases to solve problems and the ways we can make machines use them. (Kolodner, 1993)
- CBR is a recent approach to problem solving and learning (Aamodt & Plaza 1994)
- CBR is reasoning by remembering (Leake,1996)

History - USA

- ***Roger Schank, Yale University: Kognitionswissenschaftliche Arbeiten***
  - 1977: Scripts als Wissensrepräsentation (Schank, Abelson)
  - 1983: Dynamic Memory Theory, Memory Organization Packets CYRUS: Erstes implementiertes CBR-System (Kolodner)
  - 1983-1988: Weitere Systeme, z.B. : JUDGE, SWALE, CHEF
- ***Bruce Porter, Austin Texas: Arbeiten zum Konzeptlernen***
  - 1986-89: System PROTOS (Konzeptrepräsentation durch Fälle)
- ***Edwina Rissland, U. of Massachusetts: Arbeiten zur Verwendung von Fällen in der Rechtssprechung (seit 1983)***
  - 1990-92: Systeme HYPO (Ashley) und CABARET (Skalak)
- ***Jaime Carbonell & Manuela Veloso, Carnegie Mellon U.: Analogie***
  - seit 1992 Prodigy/Analogy: Fallbasierte Planung durch Analogie
- **Steigendes Interesse an CBR in USA (viele neue Forschungsgruppen)**
  - seit 1988 regelmäßige DARPA und AAAI Workshops

History - Europa

- ***Michael Richter, U. Kaiserslautern: CBR im Kontext von Expertensystemen***
  - 1988-1991 Systeme Moltke und Patdex zur techn. Diagnose
  - seit 1991 Fallbasierte Planung: Systeme Caplan/CbC, PARIS
- ***Ramon Mantaras, Enric Plaza, IIIA Blanes, Spanien: CBR im Kontext von ML***
  - 1990 Fallbasiertes Lernen zur medizinischen Diagnose
- ***Agnar Aamodt, U. Trondheim, Norwegen: CBR und Wissensakquisition***
  - 1991 System CREEK: Integration von Fällen und allgemeinem Wissen
- ***Mark Keane, Trinity College, Dublin: Kognitionswissenschaften***
  - seit 1988 Theorie des analogen Schließens
- **Seit 1991 Starkes Interesse in Europa (viele neue Forschungsgruppen)**
  - 1991 Erster Deutscher CBR Workshop (AKCBR, GWCBR)
  - 1993 Erster Europäischer CBR Workshop (EWCBR)
  - 1995 Erste Internationale CBR Konferenz (ICCBR)

CBR is a methodology

- A methodology to model human reasoning and
- A methodology for building intelligent computer systems

CBR models human reasoning

- People use previous experience to solve the current problem
  - a doctor makes diagnosis using comparison with symptoms of an other patient
  - a judge arguments with a similar case
  - a technician remembers a similar fault of the same type of machine
  - a mathematician tries to apply a similar proof for a new problem

Typical scenario - script

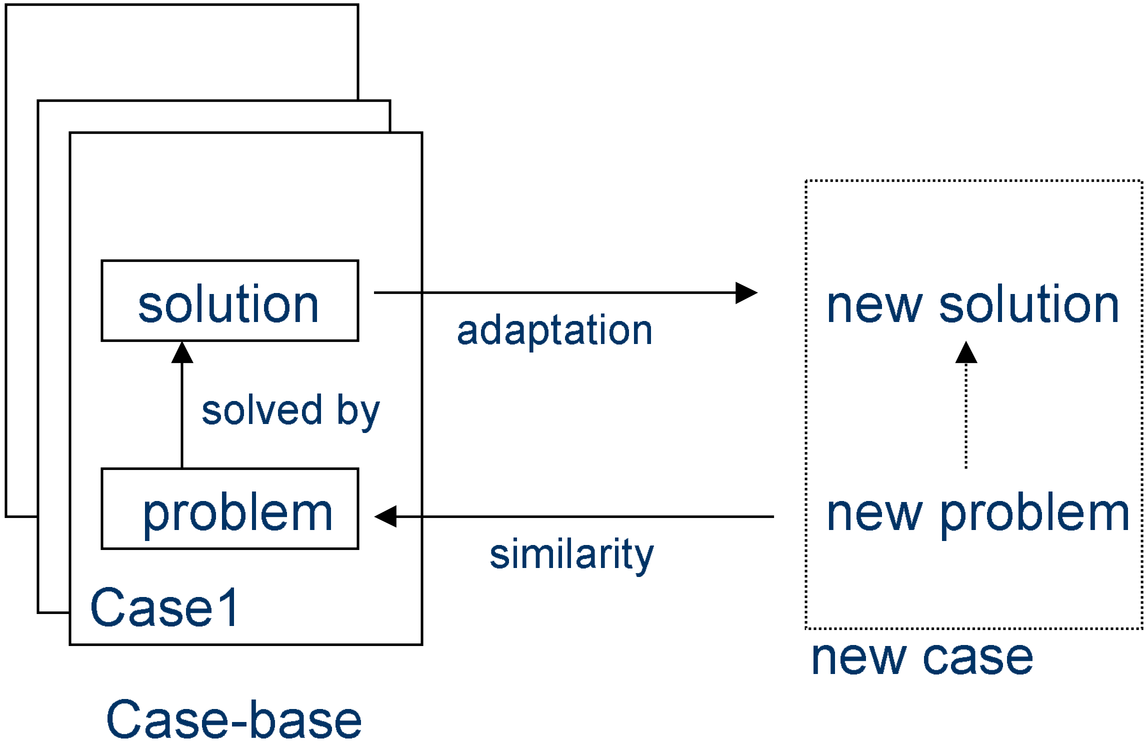
- For example - a visit to the doctor
- Script:
  - introduce yourself to receptionist
  - receptionist says “take a seat the doctor will be with you shortly”
  - sit down read a magazine for 15 mins
  - go into doctor’s room
  - get examined by doctor
  - leave with a drug prescription

Typical scenario - script

- Script for visiting a doctor will apply most times you visit a doctor
- most of it also applies when you visit other medical professionals (e. g. the dentist & the vet)
- interestingly much of it still applies when you take your car to a service

A simple CBR model

- Solve new problems by selecting previous **cases** for similar problems and adapting them to current problem



CBR system in a nutshell

- stores previous experience (**cases**) in memory

- to solve new problems:
  - retrieve similar experience (**cases**) about similar situations from the memory
  - reuse the experience (**cases**) in the context of the new situation: complete or partial reuse, or adapt according to differences
  - store new experience (**cases**) in memory (learning)

What is a Case

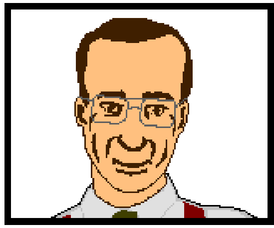
- several features describing a problem
- plus the solution or outcome
- cases can contain:
    - text, numbers, symbols, plans, multimedia,
- cases are not distilled knowledge
    - unlike rule-base systems
- cases are records of real events
- and are excellent for justifying decisions

Two types of cases

- episodic cases
  - are **real** records or **real** events
  - e.g., insurance claims, equipment faults, patients
  - can be obtained from records
  - probably require preprocessing
- prototypical cases
  - designed as **typical** examples of events
  - e.g., the typical symptoms of flue, a typical tax fraud
  - are designed by experts
  - requires knowledge elicitation

Two types of case features

Patient Ref #: *1024*
Patient Name: *John Doe*
Address: *12 Elm Street*
Next of Kin: *Jane Doe*
Photo:



Age: 53
Sex: Male
Weight: 225 lbs
Height: 5' 11"
Blood Type: A neg.
...

metadata

unindexed features

Not predictive & not used for retrieval, they provide background information to users

provide valuable contextual information and lessons learned

data

indexed features

Predictive and used for retrieval

Case vocabulary - the features that describe a case

Case representation

- Representation depends on:
  - Requirements of domain and task
  - Structure of available case data

- Flat feature-value list (like a database record)
  - Simple case structure is sometimes sufficient for problem solving
  - Easy to store and retrieve in a CBR system
  - Suitable for shallow technical diagnosis, product recommendation

Case 1:

Name:	John Doe
Address:	12 Elm Street
Age:	53
Weight:	225 lbs
Hight:	5‘ 11“
...	

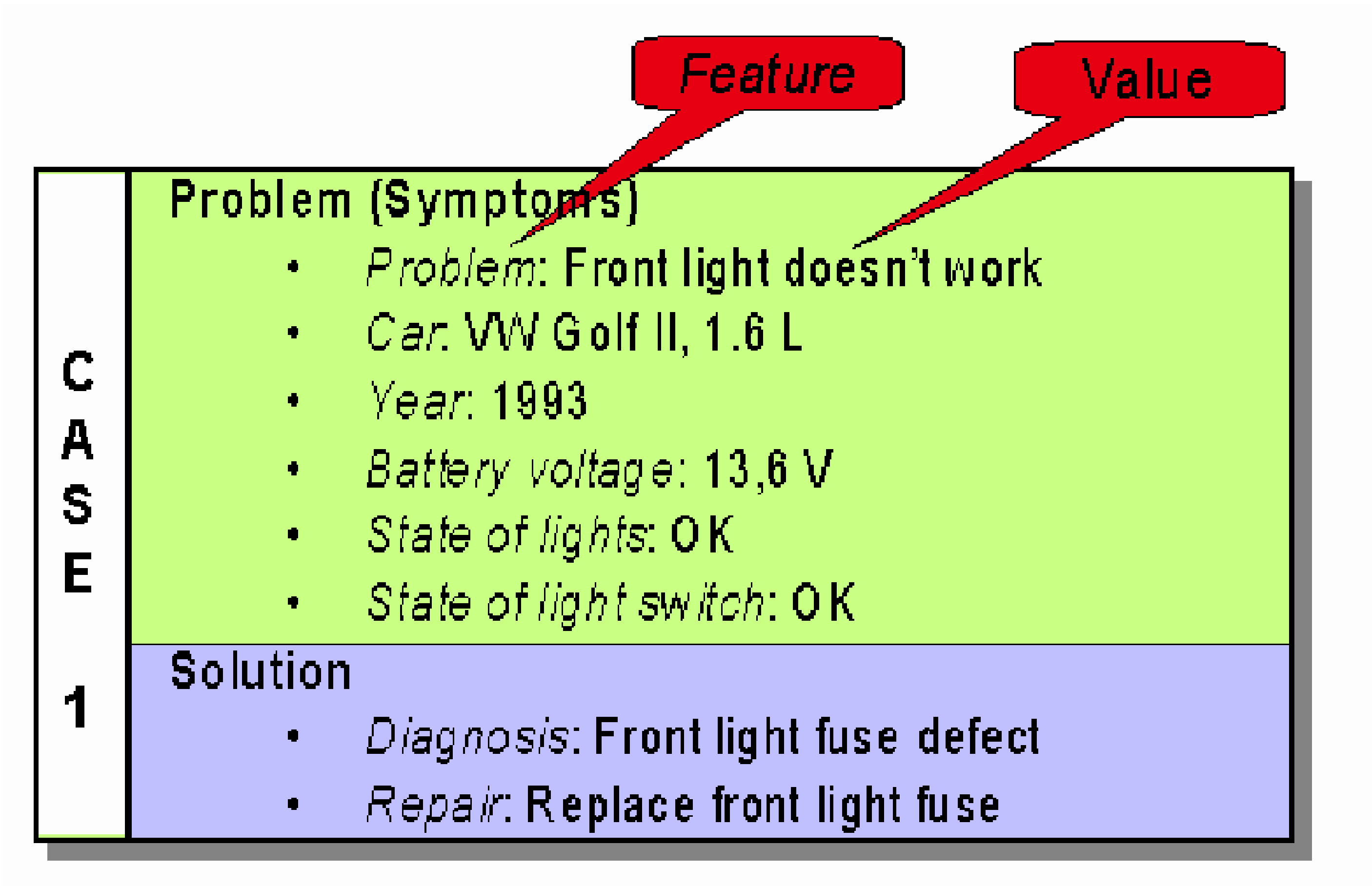
feature value

4.1.2 CBR Scenario Example - Technical diagnosis

- Simple example: Car Faults
  - Symptoms are observed (e. g. engine doesn't start) and values are measured (e. g. battery voltage = 6.3V)
  - Goal: Find the cause for the failure (e. g. battery empty) and repair strategy (e. g. charge battery)

- Case-Based Diagnosis:
  - A case describes a diagnostic situation and contains:
    - description of the symptoms
    - description of the failure and the cause
    - description of a repair strategy

CBR Scenario Example - Technical diagnosis



CBR Scenario Example - Technical diagnosis

- Each case describes one situation
- Cases are independent of each other

CASE 1

Problem (Symptoms)

- Problem: Front light doesn't work
- Car: VW Golf II, 1.6 L
- Year: 1993
- Battery voltage: 13,6 V
- State of lights: OK
- State of light switch: OK

Solution

- Diagnosis: Front light fuse defect
- Repair: Replace front light fuse

CASE 2

Problem (Symptoms)

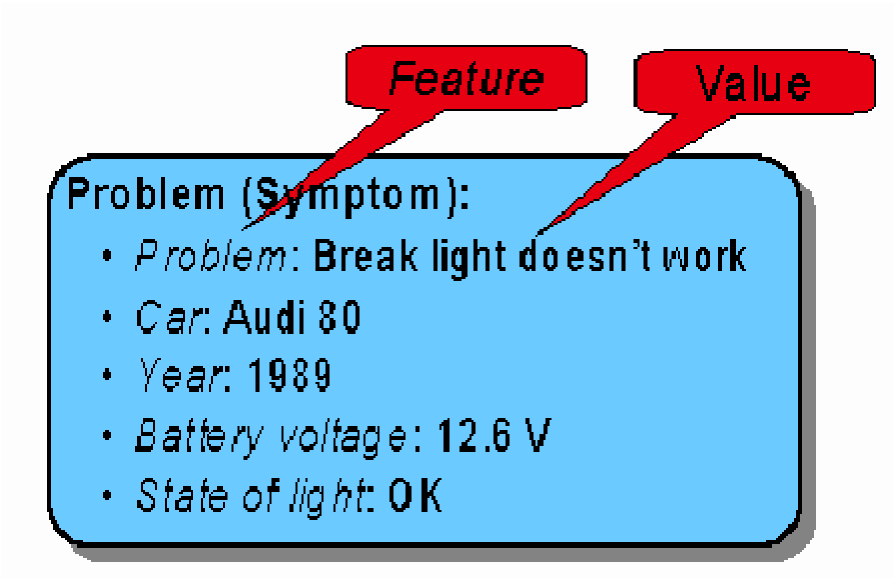
- Problem: Front light doesn't work
- Car: Audi A6
- Year: 1995
- Battery voltage : 12,9 V
- State of lights: surface damaged
- State of light switch: OK

Solution

- Diagnosis: Bulb defect
- Repair: Replace front light

Solving a diagnostic problem

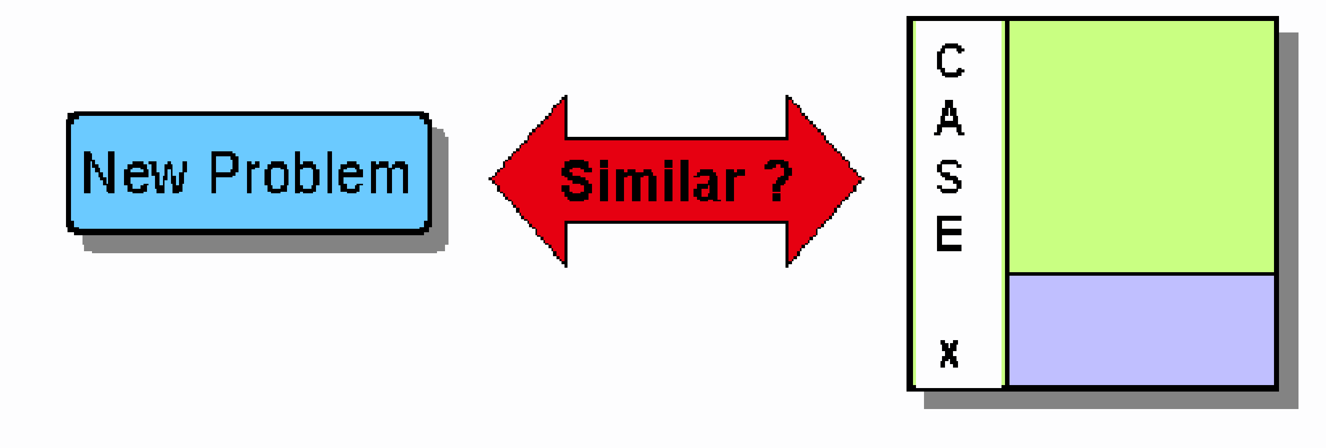
New problem:



- Make several observations about new problem
- Not all features must be known
- The new problem is a **case** without the solution part

Solving a diagnostic problem - Similarity

Compare the new problem with each case & select most similar case



- **Similarity** is the most important concept in CBR

- Purpose of similarity is to select cases that:
  - can be adapted easily to solve the current problem or/and
  - have (nearly) the same solution to the current problem

- Basic assumption:

similar problems have similar solutions

Similarity - reusability

- Degree of similarity = utility or reusability of the solution

- Goal of similarity modelling:
  - provide a good approximation
  - close to real reusability
  - and easy to compute

- Different approaches:

  k-nearest neighbour

  ML – classifiers (ID3 C4.5)

  Statistical clustering

  fuzzy sets/logic

Modeling Similarity

- Different approaches depending on case representation

- Similarity measures (metrics):
  - Functions to compare two cases:
    - sim(Case 1 ,Case 2) ⇒ [0..1]
  - Local similarity measure:
    - similarity on feature level
  - Global similarity measure:
    - similarity on case or object level
    - combines local similarity measures
    - takes care of different importance of

attributes (weights)

A similarity measure - Nearest Neighbour

- The most widely used technology in CBR
  - it is provided by the majority of CBR tools

- Algorithm:

  1. The similarity of the problem (***T*** **-** target) case to a source case (***S***) in the case-library for each (***i***) case attribute is determined by similarity function (***f***)

  2. This measure may be multiplied by a weighting factor (***w***)

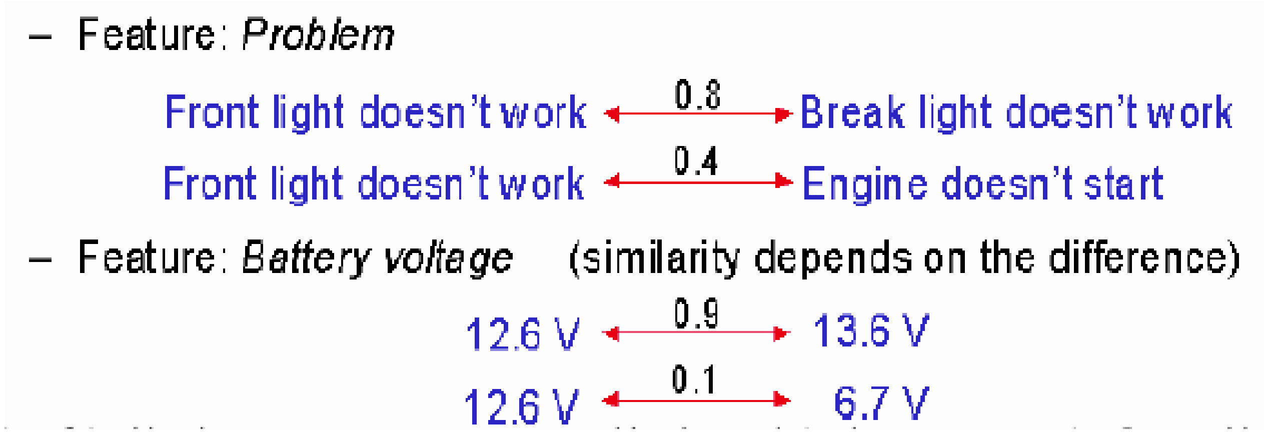
  3. Then the sum of the similarity of all attributes (***n*** -the number of attributes in each case) is calculated to provide a measure of the similarity of that case in the library to the target case

  This can be represented by the equation:

$$\text{Similarity}(T, S) = \sum_{i=1}^{n} f(T_i, S_i) \times w_i$$

Similarity

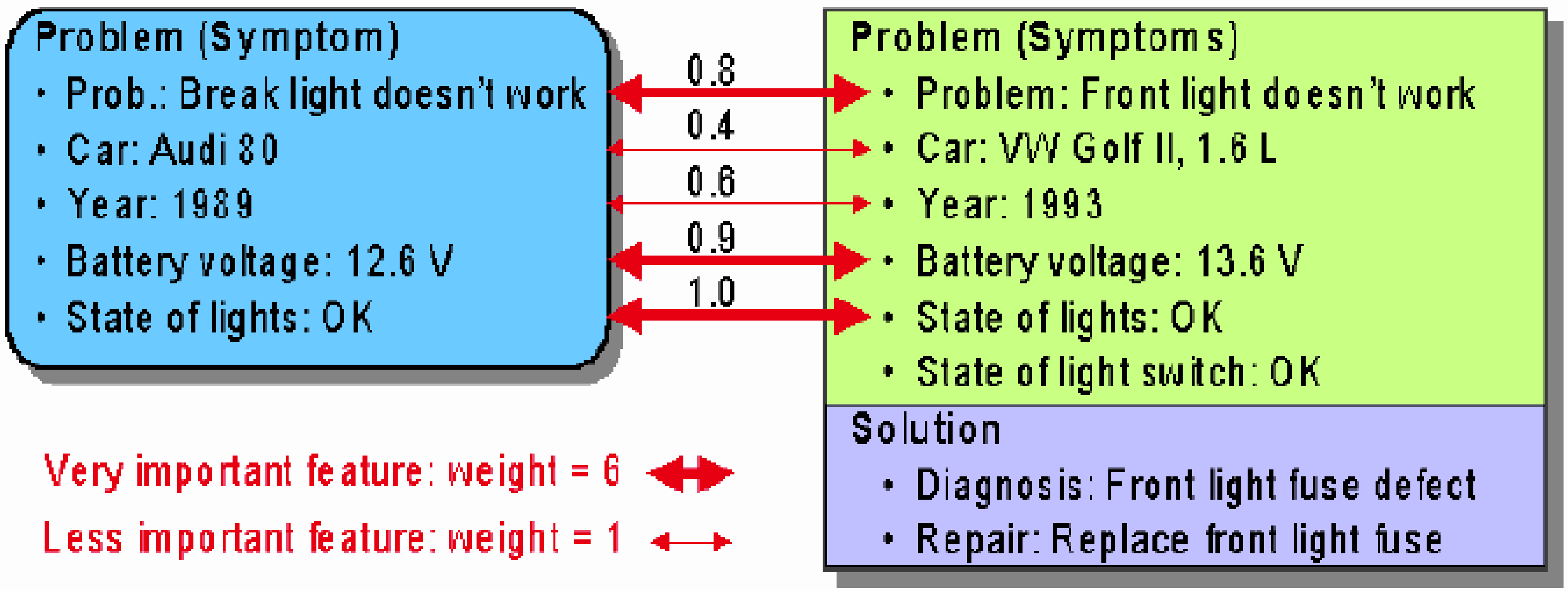
- Similarity is assessed for each feature



- Depends on the feature value
- Features can have different weights (importance)

  - High importance: Problem, Battery_ voltage, State_ of_ light
  - Low importance: Make, Model, Year, Colour

Compare new problem with case 1

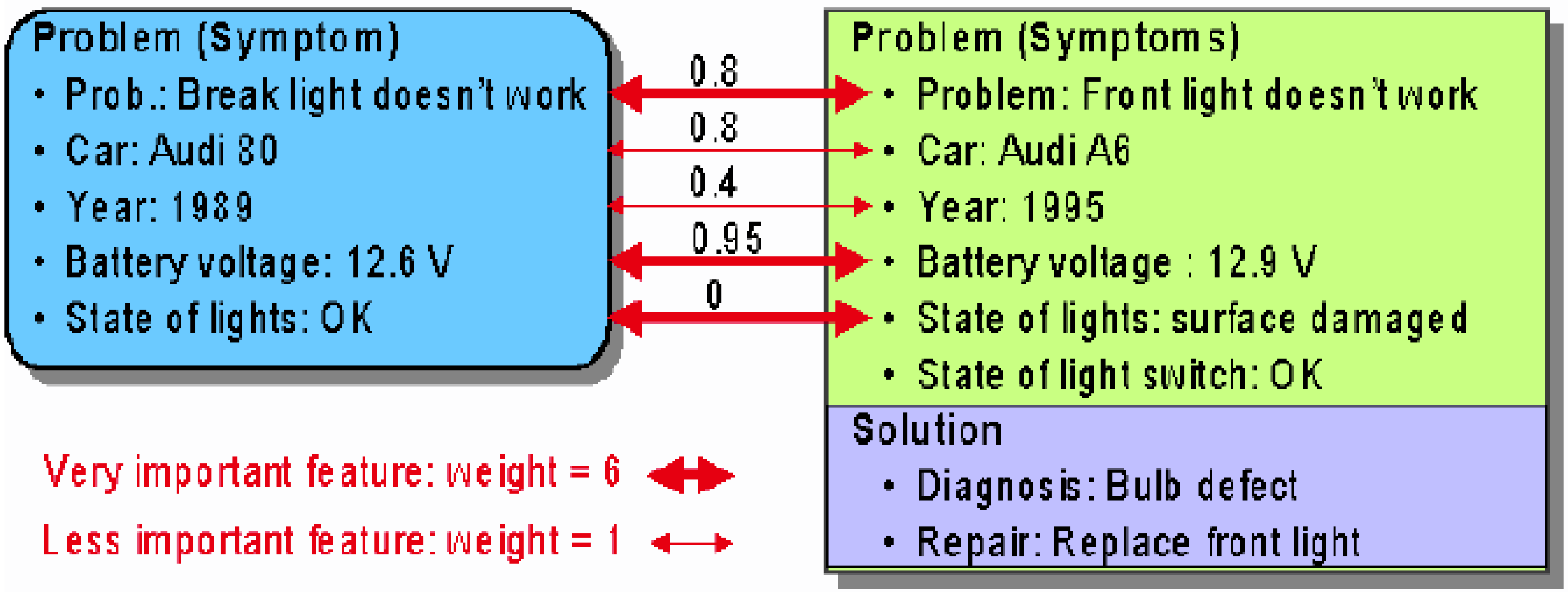


Similarity Computation by Weighted Average

similarity(new,case 1) = 1/20 * [6*0.8 + 1*0.4 + 1*0.6 + 6*0.9 + 6* 1.0] = **0.86**

Normalisation factor

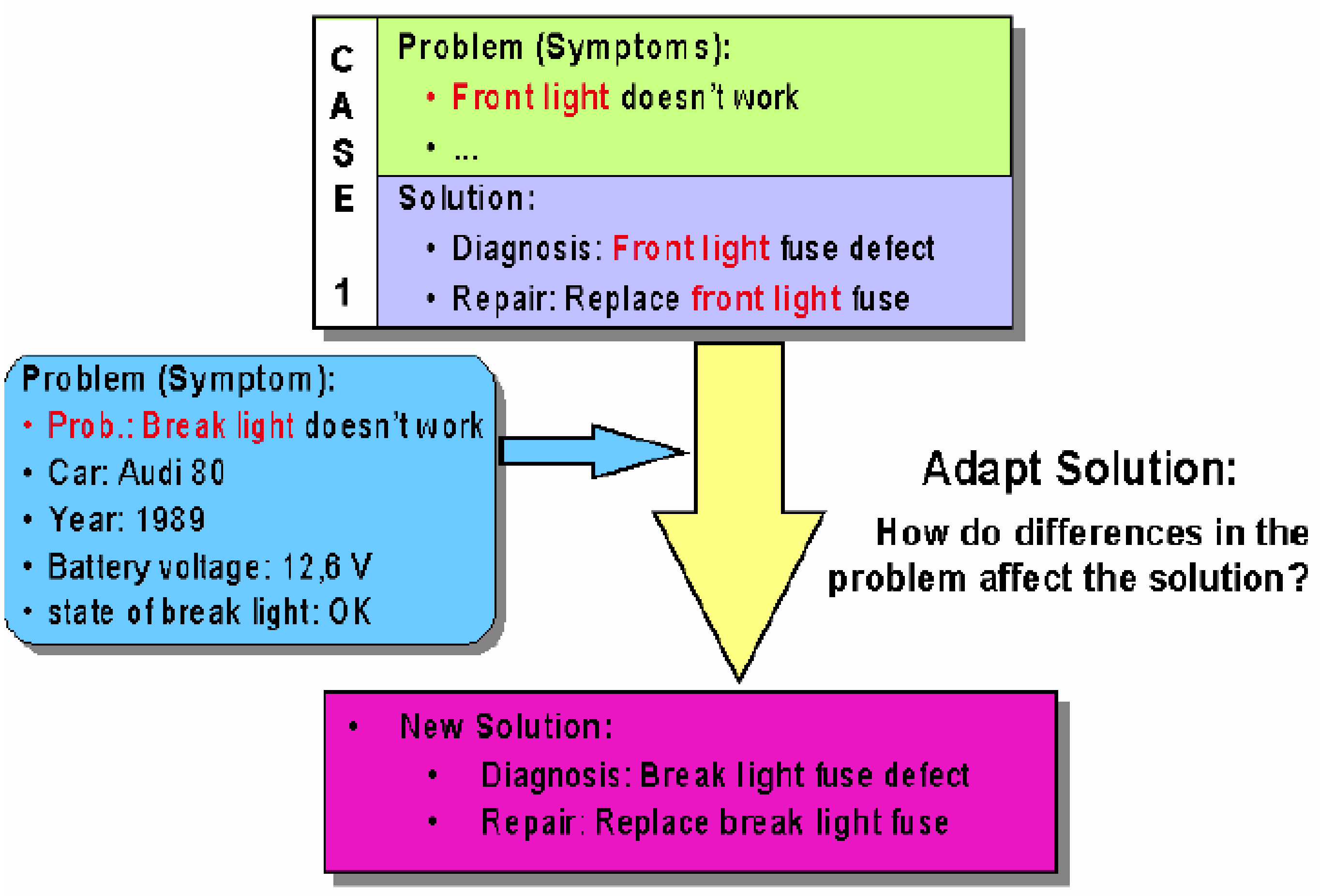
Compare new problem with case 2



Similarity Computation by Weighted Average

similarity(new,case 2) = 1/20 * [6*0.8 + 1*0.8 + 1*0.4 + 6*0.95 + 6*0] = **0.585**

Reuse the solution of Case 1



Adaptation methods

- **Null Adaptation**

- No modification of the solution: just use the solution of the closest matching problem

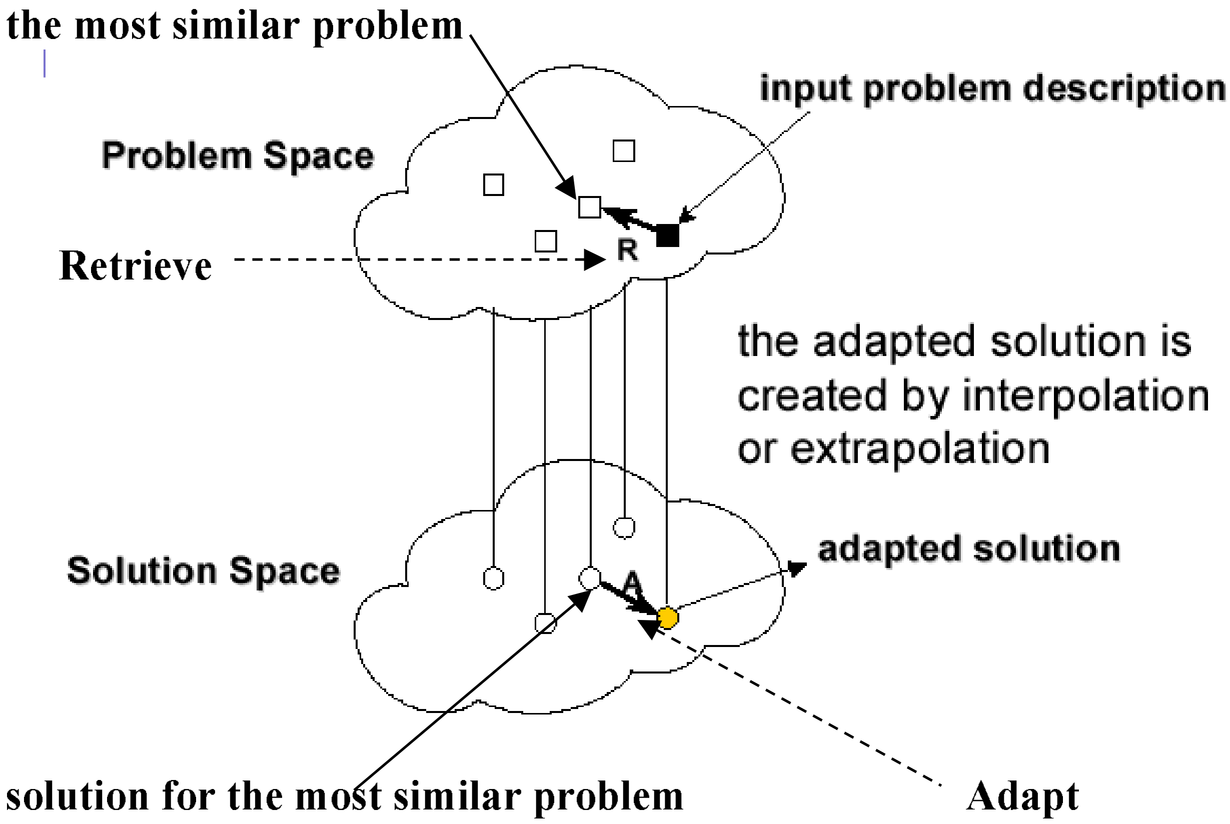
- **Manual/interactive adaptation**

- The user takes the solution of the closest matching problem using it as a basis of a new solution

- **Automatic solution adaptation**

- Transformational Analogy: Rules or operators adjust solution w.r.t. similarities and differences in the problems
- Derivational Analogy: Replay the problem solving method from the retrieved problem
- Compositional adaptation: Combine parts of several cases to form a single solution

Problem & Solution spaces



Assumptions

- 2 similar problem descriptions have similar solution descriptions
- It is easier to adapt the solution of a similar problem than the solution of a less similar problem

Store the new experience

If the diagnosis is correct:

store new case in the case base

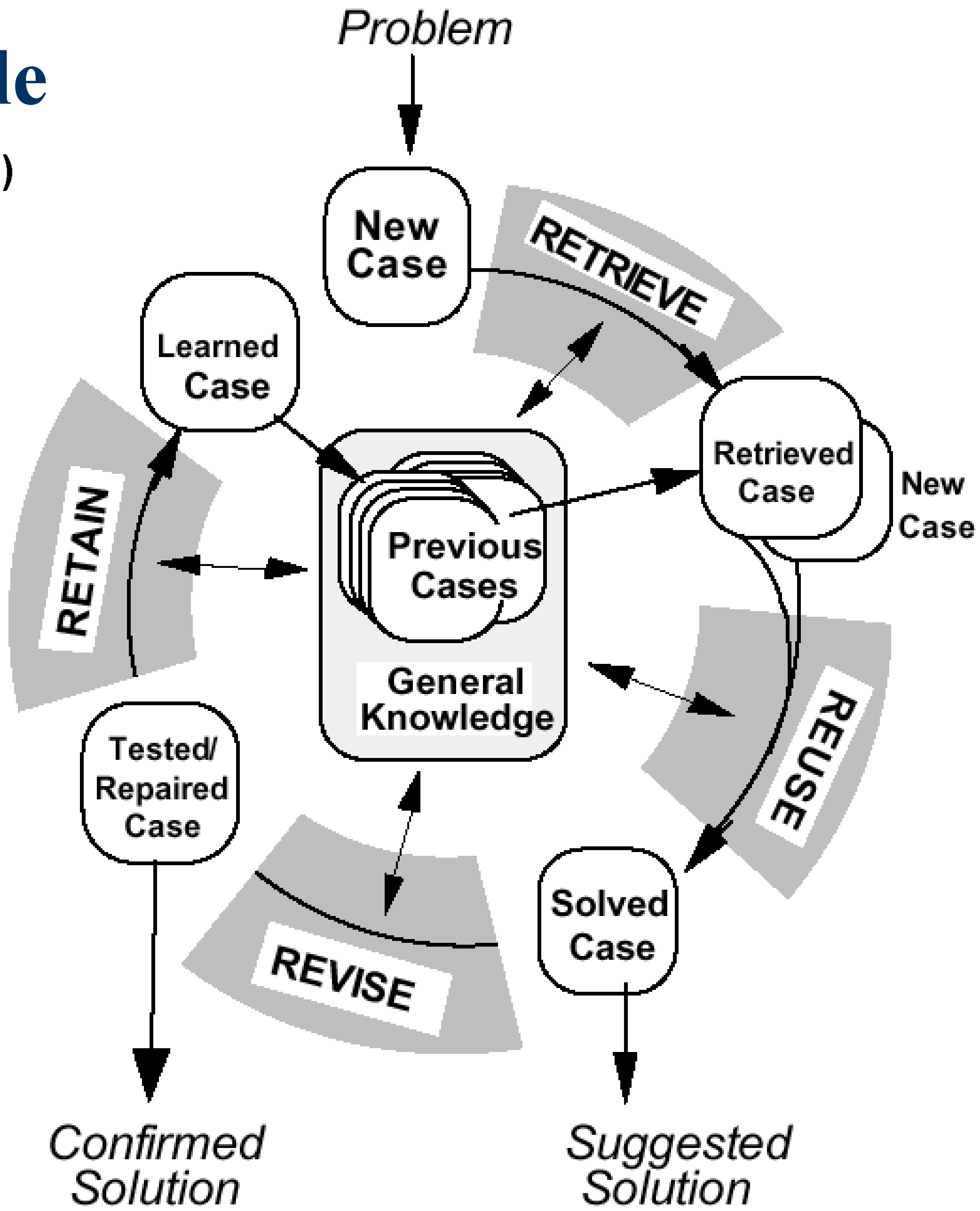
CASE 3

Problem (Symptoms):
- Problem: Break light doesn't work
- Car: Audi 80
- Year: 1989
- Battery voltage: 12.6 V
- State of break lights: OK
- light switch clicking: OK

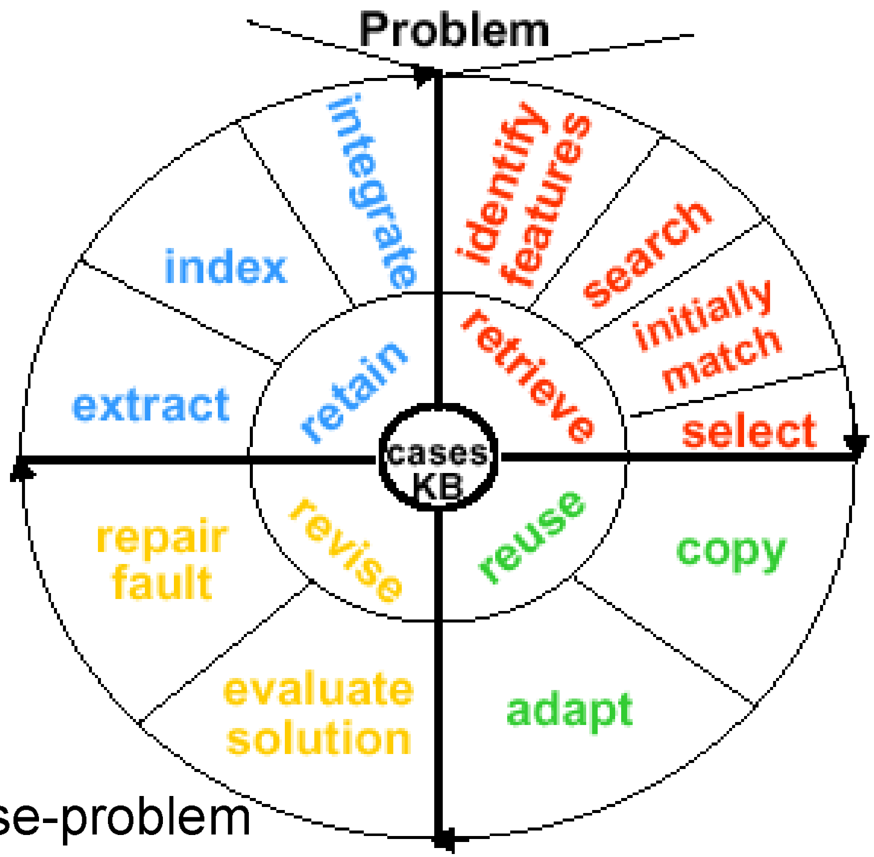
Solution:
- Diagnosis: break light fuse defect
- Repair: replace break light fuse

The CBR-cycle

(Aamodt & Plaza, 1994)



The CBR-cycle



- Process-model:

  1. Retrieve
     - find all similar cases
  2. Reuse
     - adapt to the new case-problem
  3. Revise
     - evaluate solution for the new problem
  4. Retain
     - store the acquired experience

The CBR-cycle

Example Scenario - Technical diagnosis

1. Retrieve
   - similarity
   - technical knowledge

2. Reuse
   - undertake the diagnosis of the most similar case

3. Revise
   - user-feedback (in the case of fault in diagnosis)
   - acquire more information (symptoms)

4. Retain
   - learn new cases
   - learn a better similarity measure
   - learn a better strategy to get information

4.1.3 CBR system - characteristics

- Complex knowledge acquisition is avoided
- Simple maintenance of the knowledge in the system
- High quality of retrieved solutions
- High efficiency by problem-solving
- Better reuse of existing data/knowledge
- User acceptance

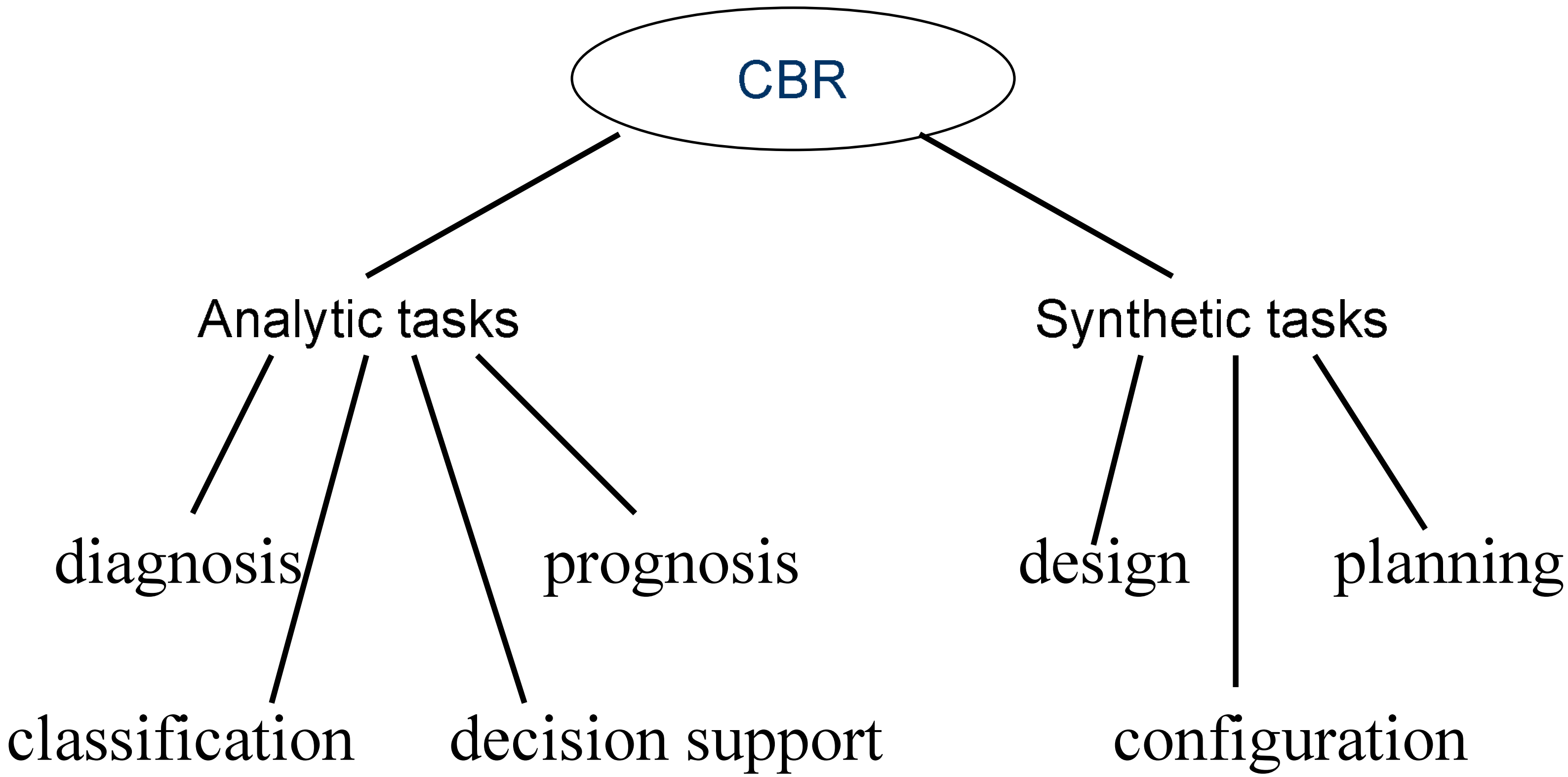
CBR is Intuitive

- this is how we routinely make decisions
- experts rely on their experience
- novices use rules and first principles
- people learn by acquiring new cases
  - experts decision making is dynamic

CBR is Simple

- easily understood by users
  - this increases acceptance
- simple to implement
- all you need are cases
- and simple software
- consequently CBR is
  - easy to sell to management
  - easy to sell to users

Problem-solving or decision-making tasks of CBR



Analytic tasks

- Charakterisierung:
  - Schwerpunkt der Anwendung liegt in der Analyse einer vorliegenden Situation
  - Situation muß hierzu i.d.R. in einer Klasse zugeordnet werden.
  - Anzahl der Klassen ist i.d.R. fest vorgegeben
  - Je nach Art der Aufgabe kommen weitere Problemlöseschritte hinzu
- Fallbasierte Systeme für analytische Aufgaben:
  - Fälle typischerweise: (Situation, Klasse)
  - Schwerpunkt liegt beim Retrieval
  - Lösungsanpassung ist oft nicht erforderlich
  - Viele Anwendungen bereits im täglichen Einsatz

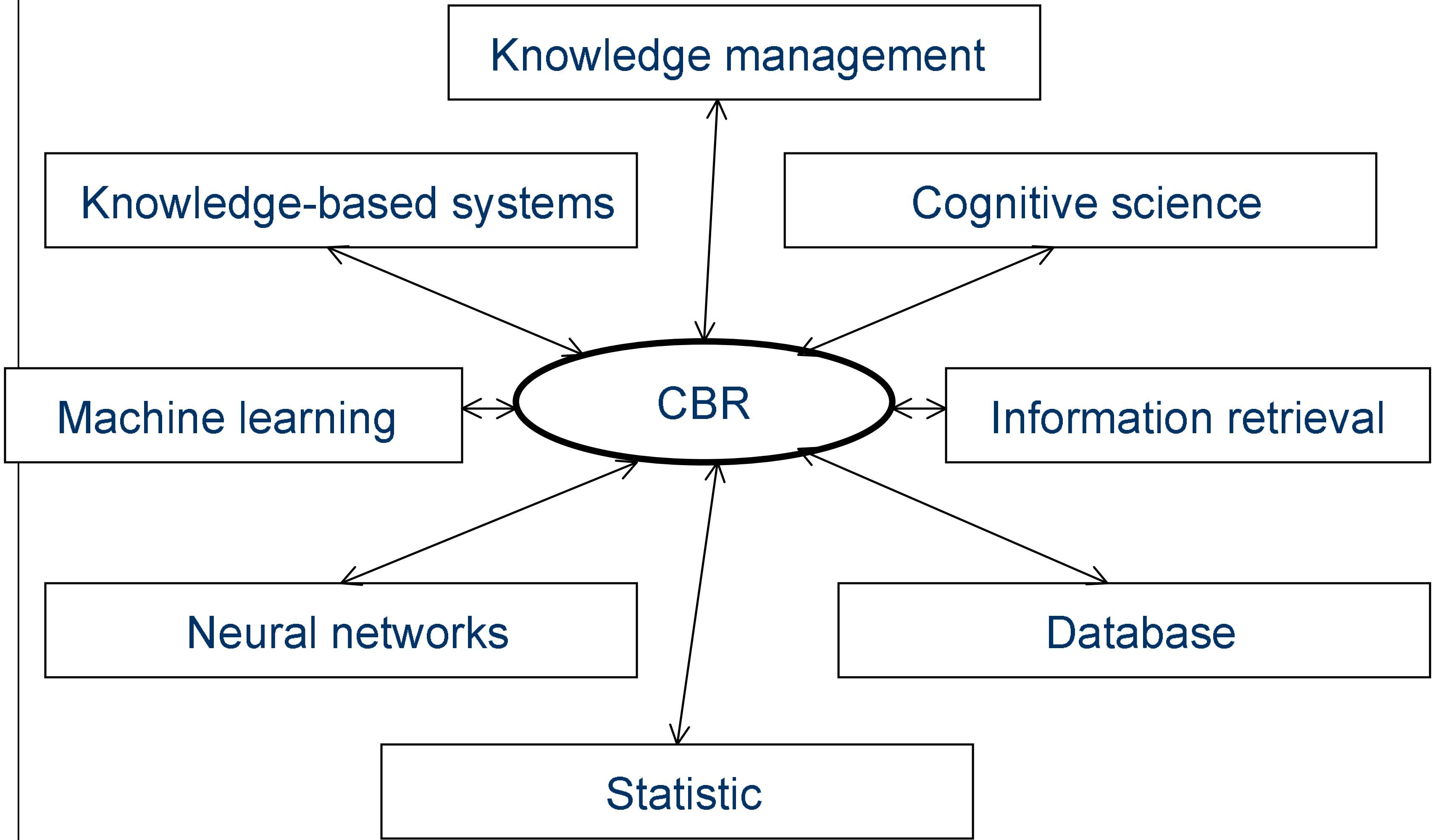
Synthetic tasks

- Charakterisierung:
  - Schwerpunkt der Anwendung liegt im Zusammensetzen einer komplexen Lösung aus einzelnen Bestandteilen
  - Problembeschreibung ist i.d.R. eine Anforderung an die Lösung
  - Lösungsraum ist i.d.R. unendlich groß
- Fallbasierte Systeme für synthetische Aufgaben:
  - Fälle typischerweise:
    - (Problem, Lösung) oder
    - (Problem, Lösungsweg)
  - Schwerpunkt liegt bei der Lösungsanpassung
  - Häufig viel allgemeines (zusätzliches) Wissen erforderlich
  - Viele Forschungsprototypen
  - Entwicklungsaufwand größer als für analytische Aufgaben
  - Anwendungen an der Schwelle zum täglichen Einsatz

When to Apply CBR?

- when a domain model is difficult or impossible to elicit
- when the system will require constant maintenance
- when records of previously successful solutions exist
- or when experts can design prototypical cases

CBR system - applicability



Who Uses CBR?

- American Express - credit card risk assessment
- Microsoft – help desks
- Barclaycard - fraud watch
- General Electric – train diagnostics, plastic fabrication
- British Airways – plane maintenance
- Daimler Chrysler – software support
- Analog – component selection
- NASA – space shuttle support
- Swiss Bank - investment management
- Deloitte Touche - fraud

4.2 CBR & KM -
Where is the Knowledge in CBR (I)

- knowledge about domain is in the case-base

- knowledge (problem-solving) in a CBR system is in 4 “containers” [Richter 95]:
  - case representation
  - retrieval algorithm
  - similarity metrics
  - adaptation methods

- knowledge (problem-solving) can be moved between containers

AIFB

CBR & KM -
Where's the Knowledge in CBR (II)

Moving knowledge between containers

- example 1
  - imagine a CBR system that **just retrieves** any old case **then adapts** it to a good solution
  - all the knowledge is in the **adaptation** component

- example 2
  - imagine a CBR system that uses a complex highly structured case representation to **always retrieve** good solutions
  - most knowledge is in the representation container, some in **retrieval and similarity** and no knowledge in adaptation

CBR & Knowledge Management

Framework for comparison (I)

- Goal is the same:
  - to *capture* and *reuse* experience or knowledge (to improve and support the overall business strategy of an organisation)

CBR & Knowledge Management

Framework for comparison (II)

- CBR vs. KM:
  - less emphasis on knowledge creation aspects
    - cases are assumed to exist
  - very simple knowledge representation
    - feature-value pairs (suitable for databases)
  - strong support for knowledge retrieval and adaptation
    - contextual retrieval is supported
  - suitable for processing noise-data
    - similarity is the most important concept in CBR
  - applicable only in domains in which similar problems occur
  - modelling background knowledge in cases not possible

CBR & Knowledge Management

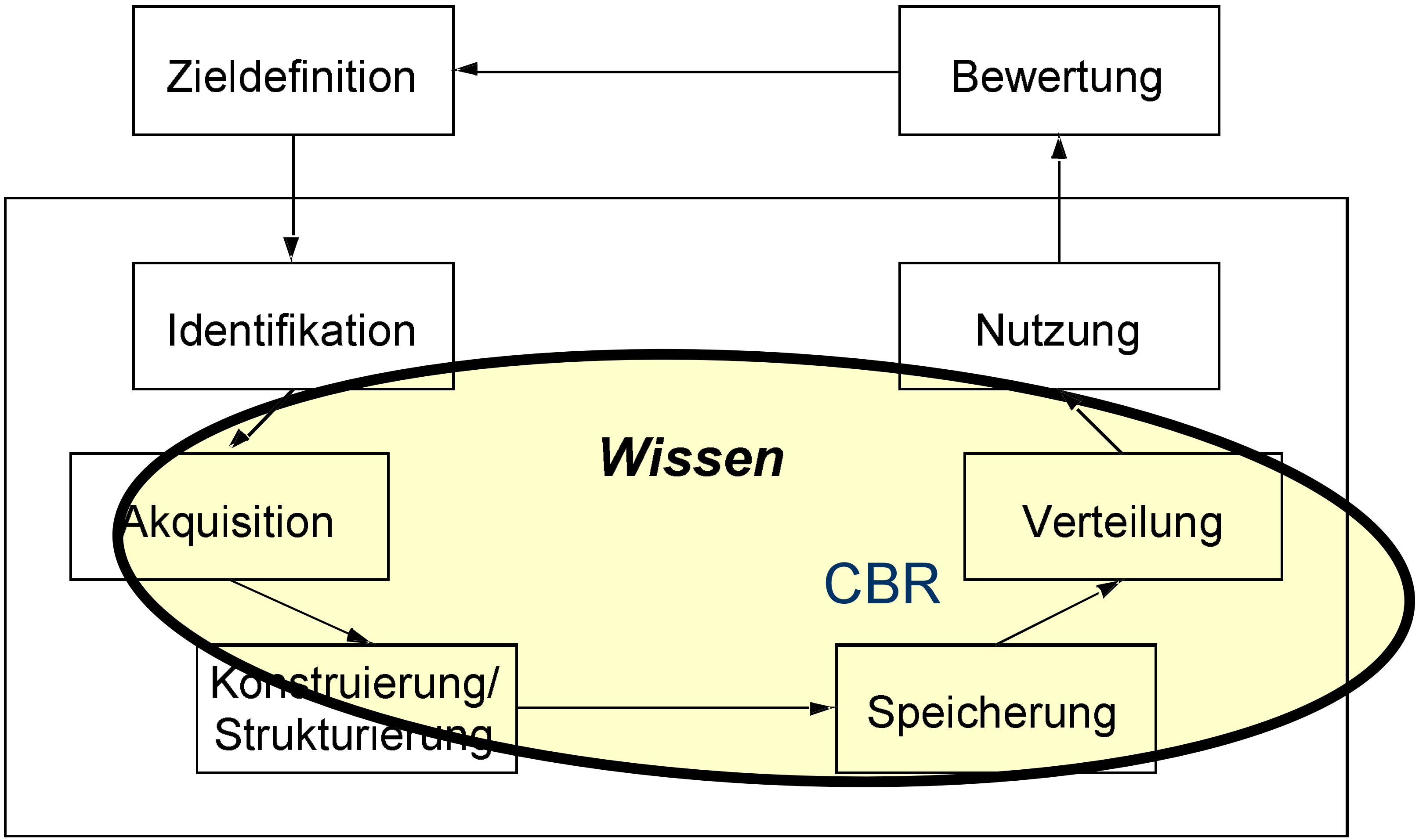
Framework for comparison (III)

- KM vs. CBR:
  - strong support for domain description (background knowledge)
    - e.g. ontology-based KM
  - (business) process-oriented
  - strong support for a global management process in the organisation
    - knowledge management covers also non-IT aspects
  - less emphasis on knowledge sharing aspects
    - it is assumed that for a given problem useful knowledge could be found
    - knowledge should be *accessed* based on the understanding of what the knowledge seeker *needs* or *wants* to know

4.2.1. CBR subsystem for storing/retrieving knowledge

- **Primäre Aufgaben des Wissensmanagements**

vgl. [Probst et al. 1997]



CBR subsystem for storing/retrieving knowledge

Example: The World Bank, NEC

Aquisition/Structuring

- KM: Knowledge acquisition bottleneck
- CBR: Complex knowledge acquisition is avoided - knowledge is stored in the form of cases
  - but not background knowledge

Delivering/Sharing

- KM: How to find useful knowledge
- CBR: Case has several features which describe a problem plus the solution or outcome
  - problem description is used for retrieval

CBR subsystem for storing/retrieving knowledge

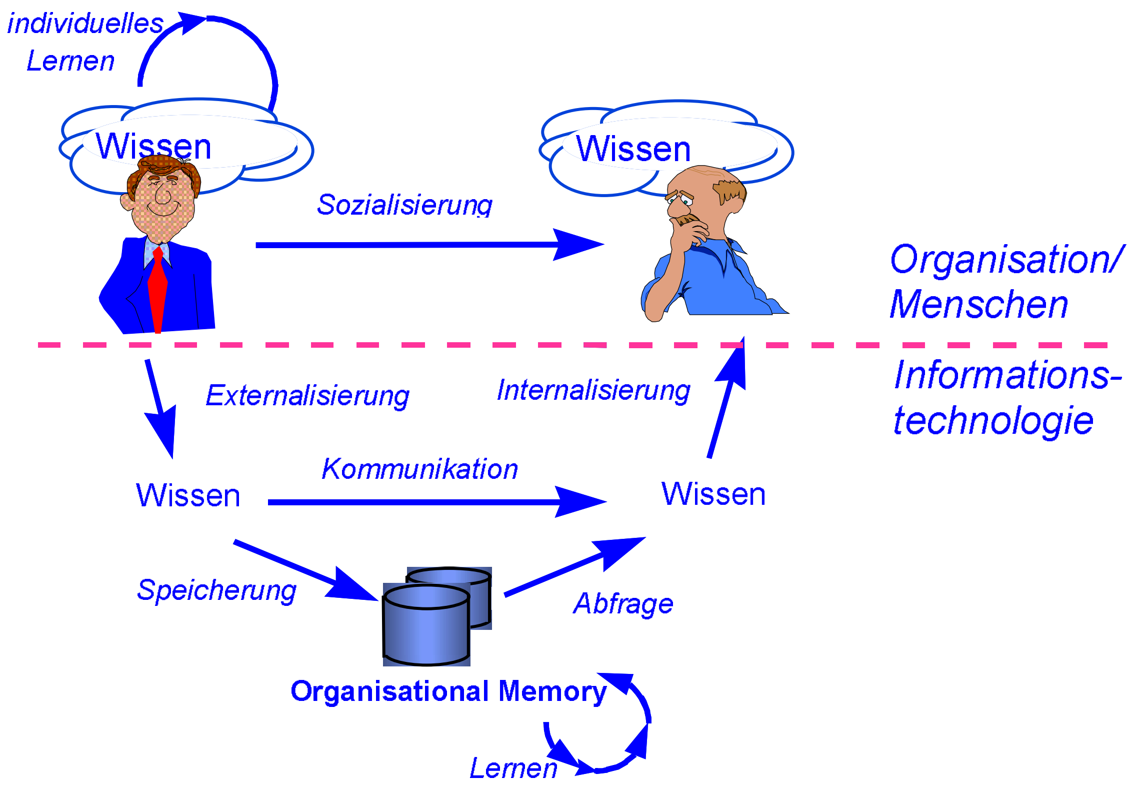
Benefits:

- storing/access to the large portion of tacit or unstructured knowledge (stored in cases)
  - e.g. staff decisions and judgements in taking certain actions, reasons for successes and failures of projects
  - lessons learned and good practices
- integrated description problem-solution
  - case contains description of problem and **knowledge** how to resolve it
- by accessing: retrieve only relevant and meaningful information
  - users should be able to articulate exactly what they want
  - knowledge sharing could be empowered by relevance-based retrieval, case reuse, and learning

4.2.2 CBR system for Organisational learning

Learning in an organisation

„Knowledge is not created by people sitting and staring out their window. *Ideas* may be created [during such moments], but knowledge in the sense of capacity for effective action is created by working teams.“ - P. Senge



Organisational learning

• Learning is a dynamic concept and it emphasizes the continually changing nature of organizations

- organizations want to be more adaptable to change

• **Organisational learning (OL)** is the process how companies

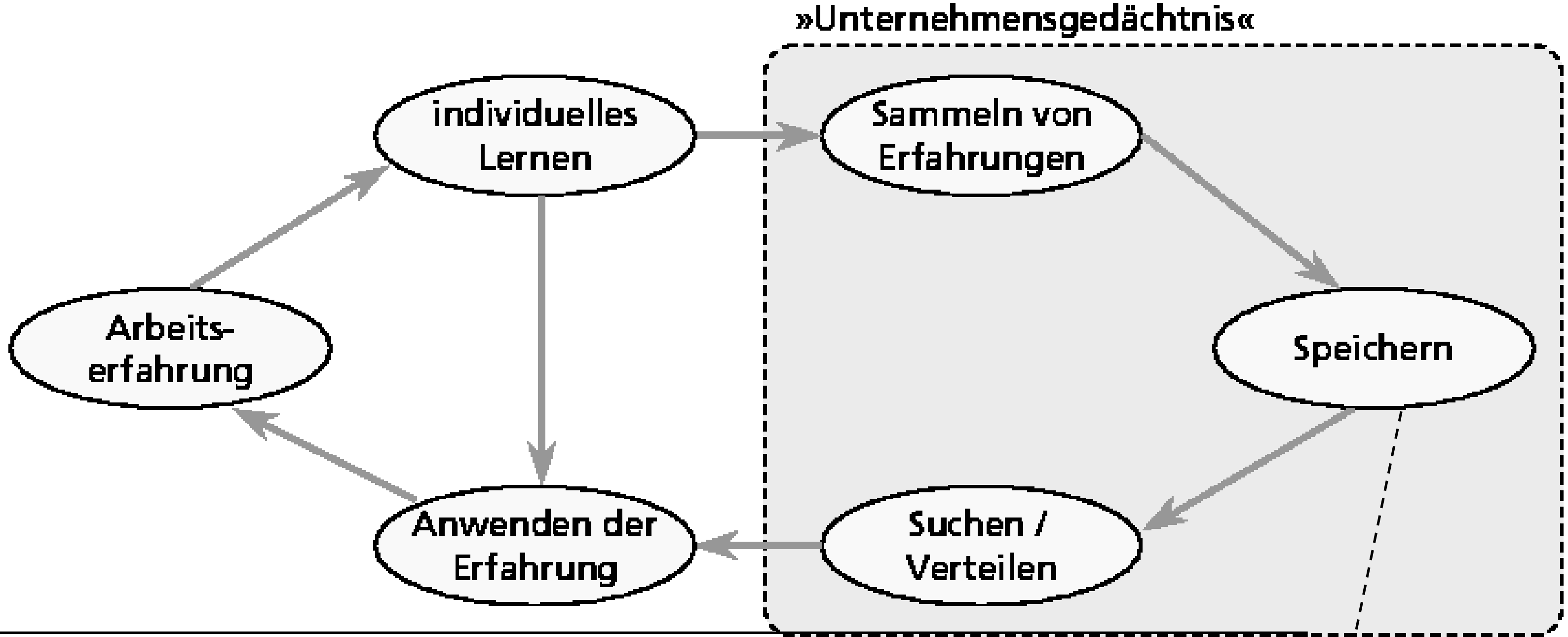
- build, supplement, and organize knowledge and routines
- around their activities and within their cultures and
- adapt and develop organizational efficiency
- by improving the use of the broad skills of their workforces

• A **learning organization** is a company that purposefully constructs structures and strategies so as to enhance and maximize organizational learning

• **Organizational learning** *anchors* the process of **knowledge management** in the core phenomenon of how new knowledge is generated - by working teams (Senge, 1999)

Organisational learning

Nach Heist et al., 1998



- **experience-, Lesson Learned- archives** / Best practice databases
  - document management
- **shared case bases**
  - application specific problem/solutions archives
- expert systems

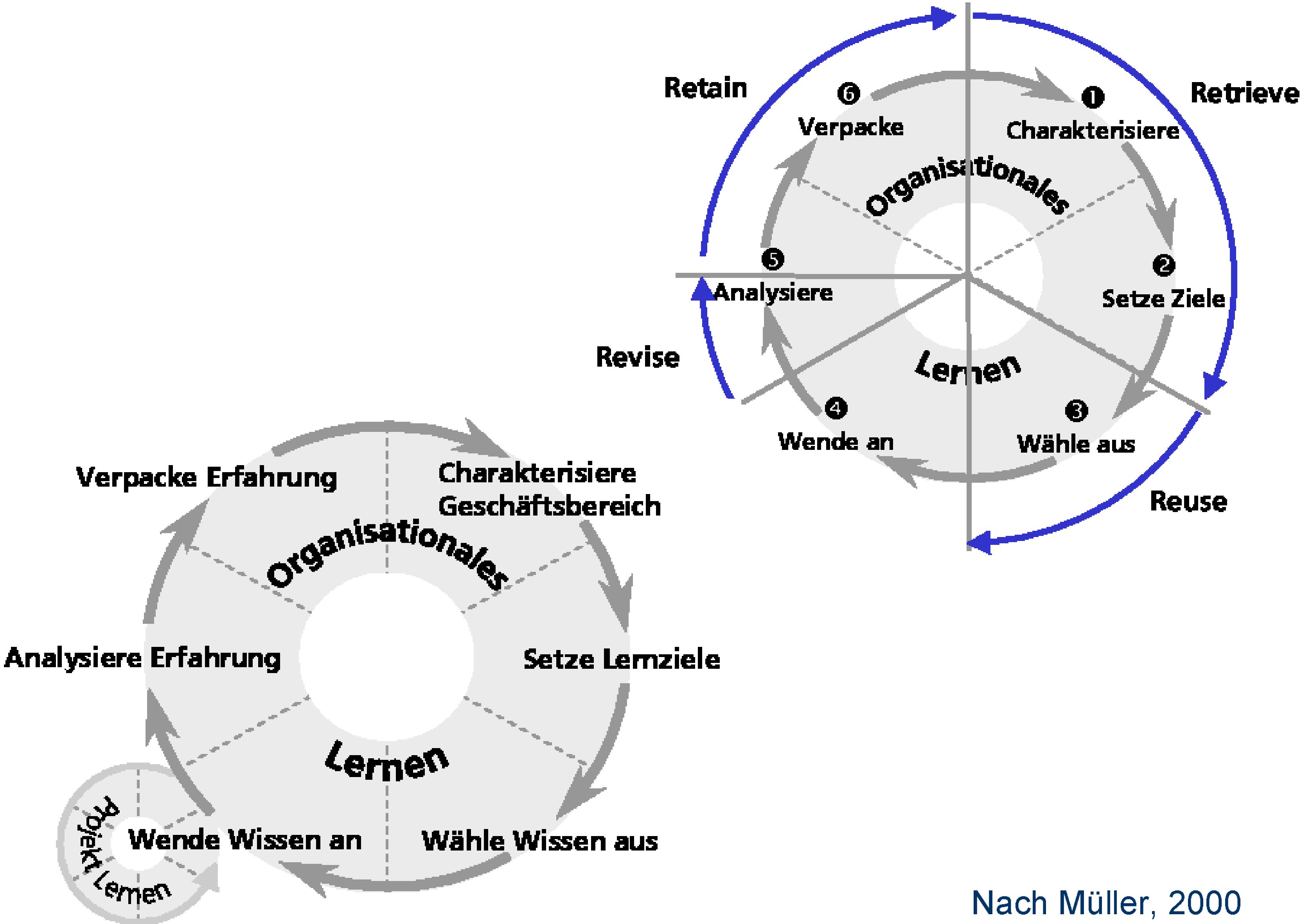
Nach Kühn/Abecker, 1998

CBR system for OL - Experience factory

- The experience factory approach aims to
  - establish an **organizational infrastructure** to
  - facilitate systematic and continuous **organizational learning**
  - through the **sharing and reuse of experiences** in software engineering

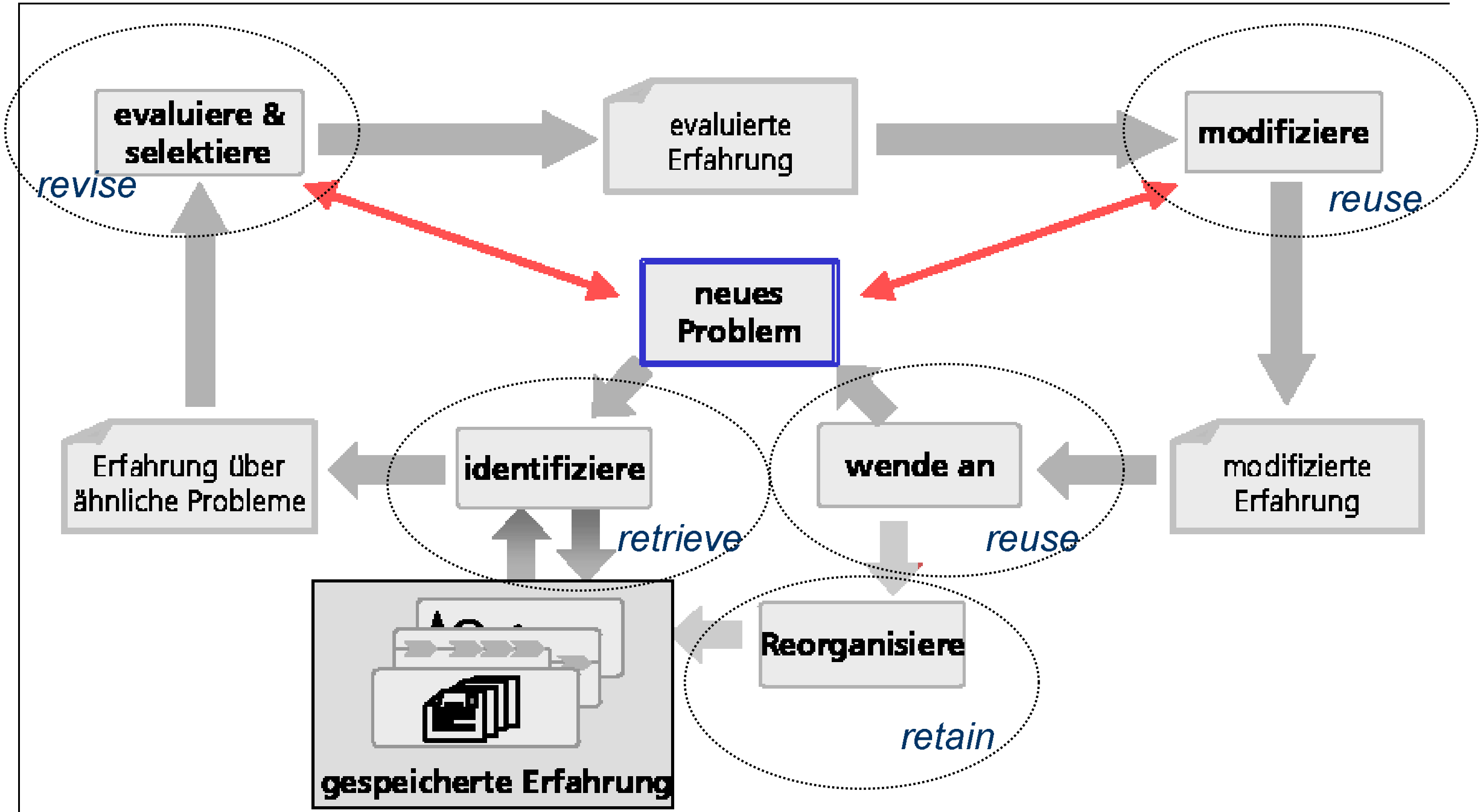
- It involves setting up a group separate to the development teams, called the experience factory, which is responsible for
  - *characterising* the problem domain and *setting* goals of learning
  - *selecting* knowledge that could be applied and *applying*
  - *collecting* and *validating* experiences from development projects,
  - *packaging* the experiences by building empirical models and structuring informal knowledge, and
  - *spreading* experience packages into development projects.

Organisational learning and CBR



Nach Müller, 2000

Experience factory - reuse of experience



Experience factory - benefits

- Support for incremental, continuous learning
- Support for maintenance of experience
- Learning from examples (natural approach)
- Context sensitive retrieval
- Support for evaluation of experience -quality, -applicability, -benefit of use
- Structuring and storage of diverse kinds of experience
- Retrieval based on incomplete information
- Retrieval of similar experience