IT2702 - Høst 2005, Leksjon 8

- **Modell-basert resonnering**
- **Case-basert resonnering**
- **Kombinerte resonneringsmetoder**

Model-Based Reasoning

- **Reasoning: Based on ”deeper” knowledge than rules**

  Typical models:
  - causal
  - functional
  - behaviourial
  -> a combination of several submodels

- **Representation**

  Different relations than rule-based ”if-then” relations:
  - taxonomical (”has-subclass”, ”has-instance”)
  - ”has-part”
  -”causes”
  - ...

  Often multiple relations combined!

Case-Based Reasoning

Motivation:

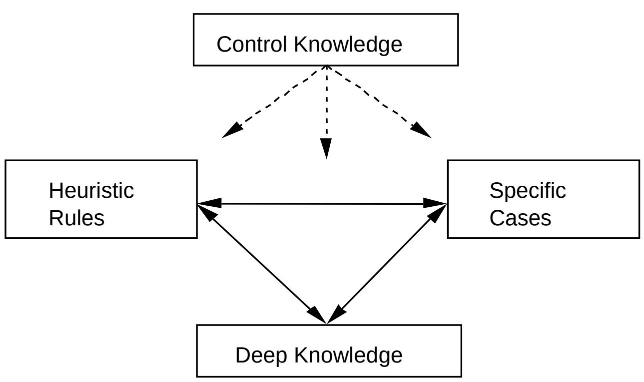
- From **cognitive science:**

  A theory of understanding,
  problem solving and learning
  in human beings.

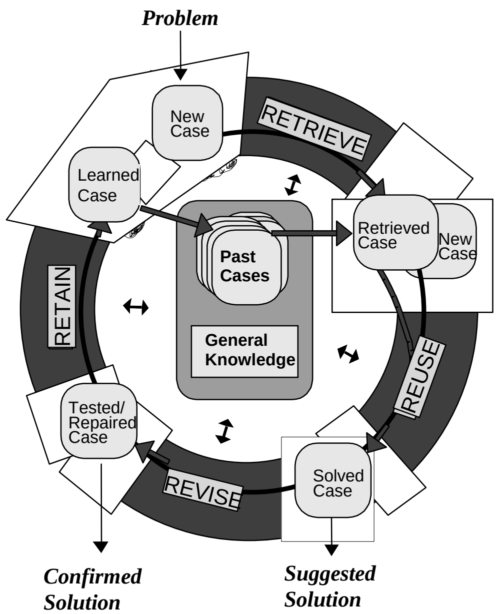
- From **knowledge-based systems:**

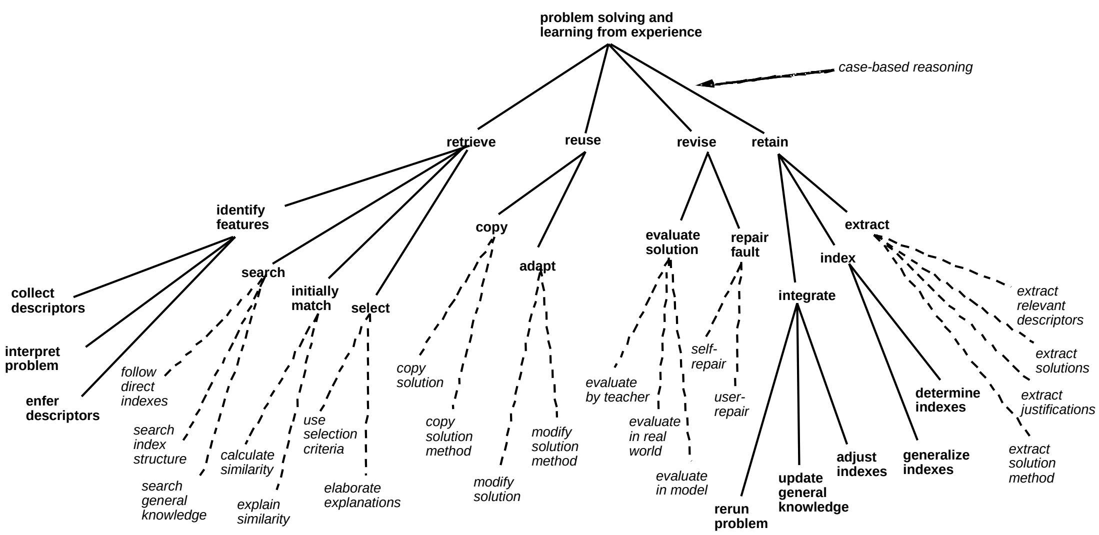
  Deficiency of purely generalization-based
  methods for intelligent computer
  programs.

KBS - Development trends



The CBR Cycle





Case-based approaches

- Instance-based reasoning/learning
- Memory-based   reasoning/learning
- Case-based reasoning/learning (typical)
- Analogical reasoning/learning

Instance-based methods

- Motivated by classical machine learning research
- Addresses classification tasks
- A concept (class) is defined by its set of exemplars:
  Concept space = Instance space + Similarity metric
- Representation is attribute-value pairs
- Knowledge-poor method
- 'IBL' framework (Kibler&Aha) contains
  - Similarity function
  - Classification function
  - Concept decsription updater

Memory-Based Reasoning

- Motivated by parallel computer architectures
- Adds parallelity to instance-based approach
- Computes distance between input and all exisiting instances
- Best match algorithm takes constant time
- Syntax-based: Trades knowledge for 'brute' power

RETRIEVE:

1. Count feature occurences; this determines relevant features.
2. Generate similarity metric from counts
3. Calculate dissimilarities
4. Find best matches

Analogy-based methods

- Motivated by psychological research
- Reuse of cross-domain cases
- Emphasis on Reuse, not Retrieval
- Computationally complex problem

Example

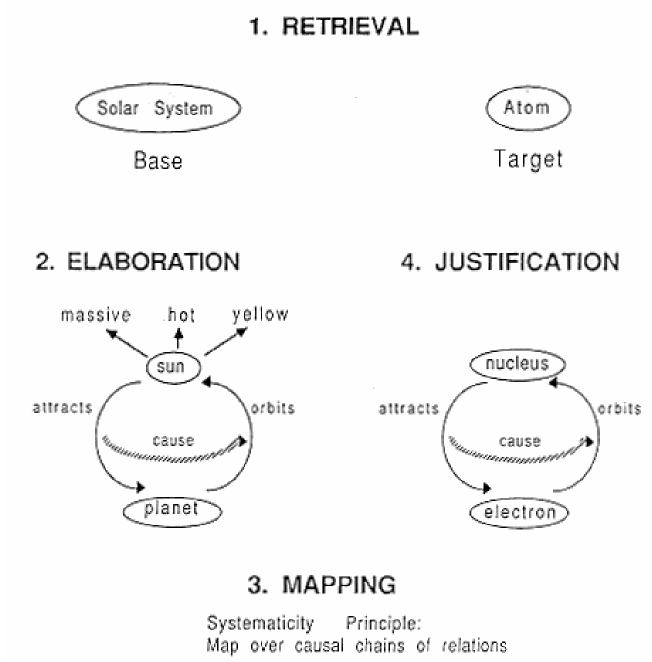
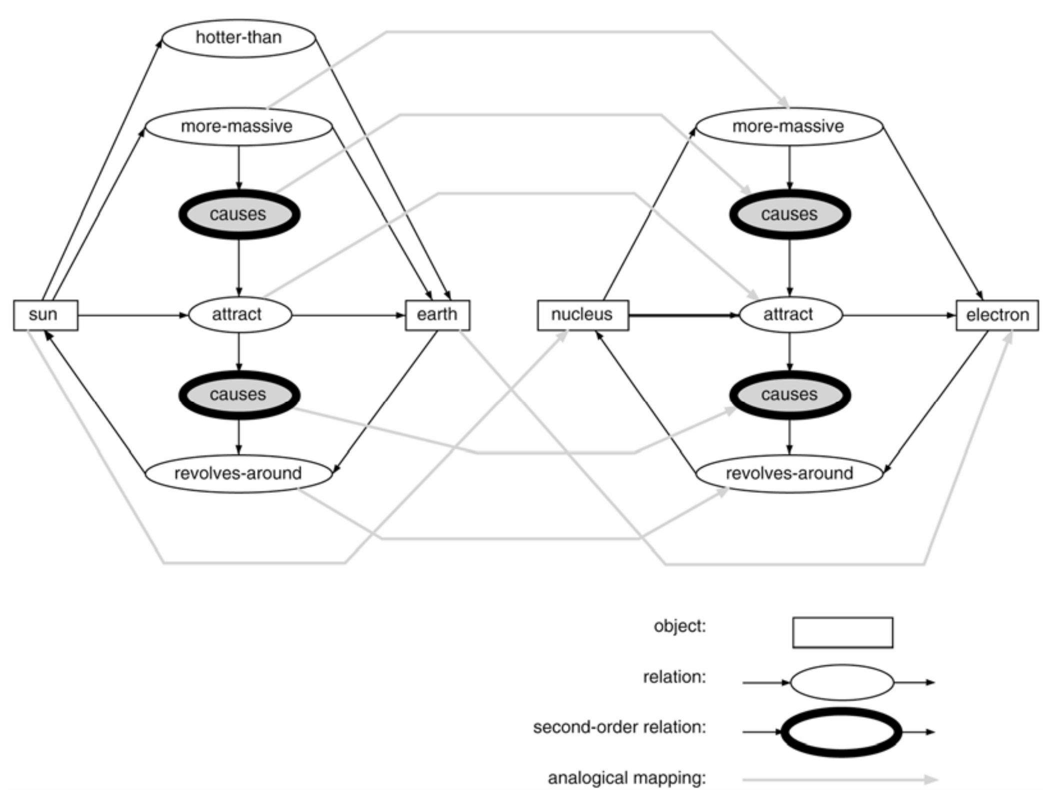
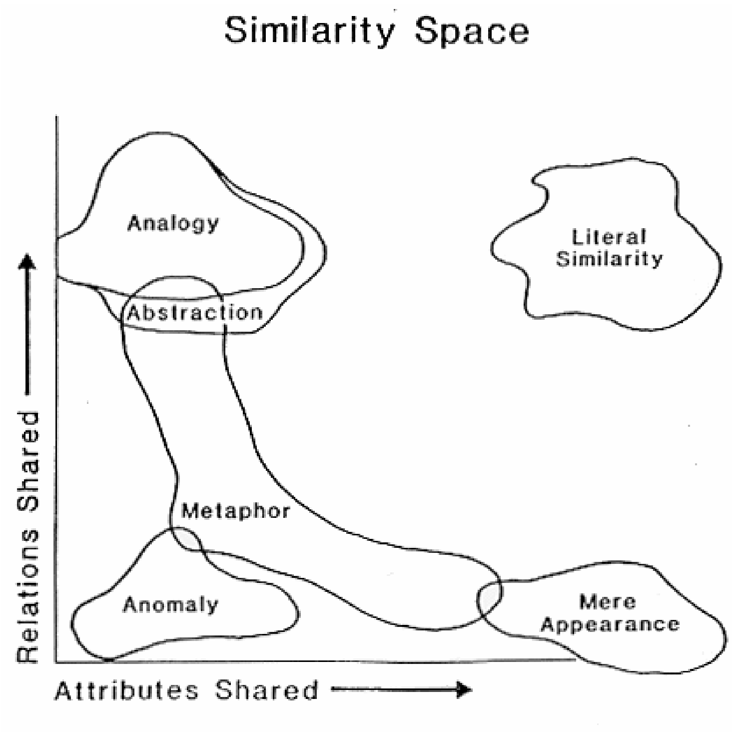


Fig. 3-3. Gentner: The atom is like our solar system.

Figure 9.19: An analogical mapping.



Relations vs. attributes



Case-based methods

(in a 'typical' sense)

- Motivated by learning for problem solving,
  rather than for general concept definitions.

- Typically uses some background knowledge
  in its Retrieval, Reuse, and/or Learning methods.

- A range of different approaches distinguished by

  - task and domain type addressed
  - memory organization (case storage, indexes)
  - case retrieval, reuse, and learning method

CBR - History

excerpt

Theoretical:

Schank/Abelson 77:	Scripts
Rissland 80:	Precedents in legal reasoning
Schank 82:	Dynamic memory, MOPs
Carbonell 83:	Transform./Derivational analogy
Kolodner 83:	Episodic memory
Schank 86:	Explanation patterns
Richter 90:	Similarity and uncertainty

Some systems:

Lebowitz 80:	IPP	- nat. language
Kolodner 83:	CYRUS	- info retrieval
Simpson 85:	MEDIATOR	- negotiation
Hammon 86:	CHEF	- cooking planning
Sycara 87:	PERSUADER	- negotiation
Ashley/Rissland 87:	HYPO	- law interpret.
Bareiss/Porter 88:	PROTOS	- medicaldiagnosis
Koton 89:	CASEY	- medical diagnosis
Goel/Chandra 89:	KRITIK	- mechanical design
Hinrichs/Kolodner 91:	JULIA	- meal planning
Aamodt 91:	CREEK	- mud diagnosis
Leake/Schank 92:	ACCEPTER	- explaining
Lopez/Plaza 93:	BOLERO	- medical diagnosis
Althoff/Wess/Richter 93 :	PATDEX	- technical diagnosis
Oehlmann/Sleeman94:	IULIAN	- discovery, planning
Esprit-project -95	INRECA	- CBR and induction

Transformational and Derivational ”analogy”
(J. Carbonell 83)

- Transformational

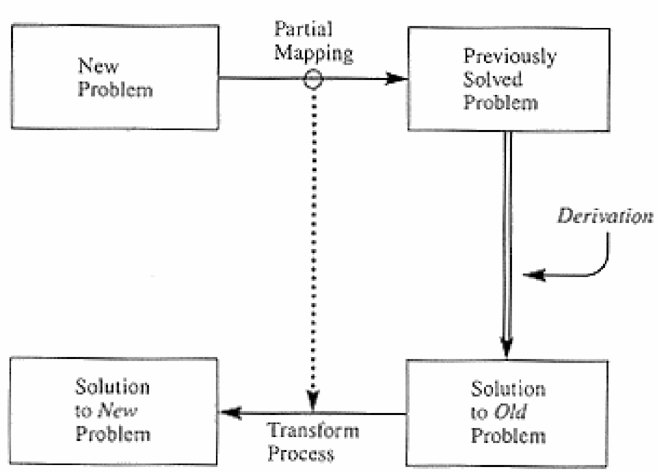
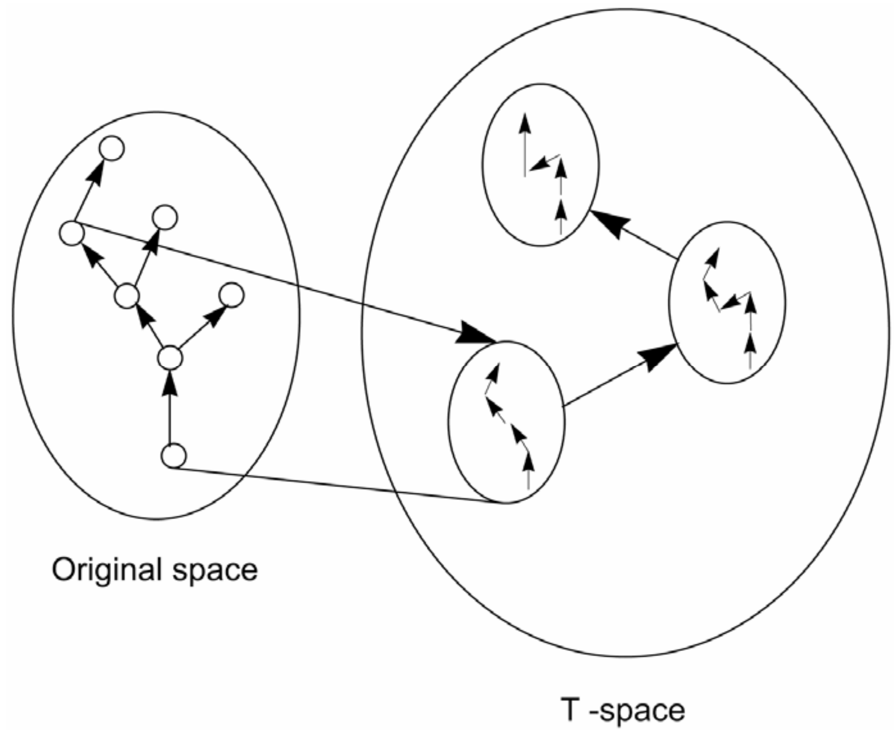
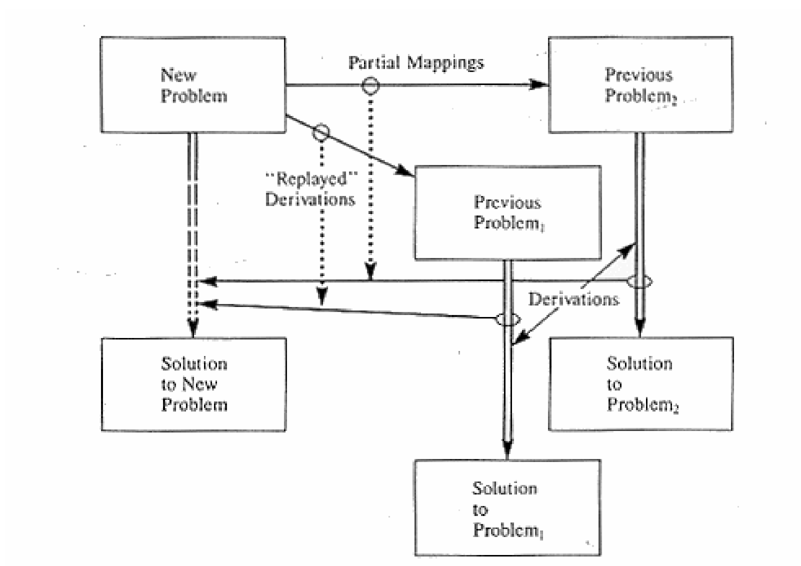


Figure 7.17: Transformational analogy, adapted from Carbonell (1983).



- Derivational



Problem areas

- Memory organization
  - case structure
  - index structure
  - integration of general domain knowledge
- Retrieval
  - use of indexes
  - feature relevance
  - similarity assessment
  - use of general knowledge
  - use of previous cases
- Reuse
  - transfer of solution
  - adaptation of solution
  - transfer (and adaptation) of solution method
- Learning
  - feature extraction
  - as separate cases vs. splitted up
  - index learning
  - generalization
  - forgetting

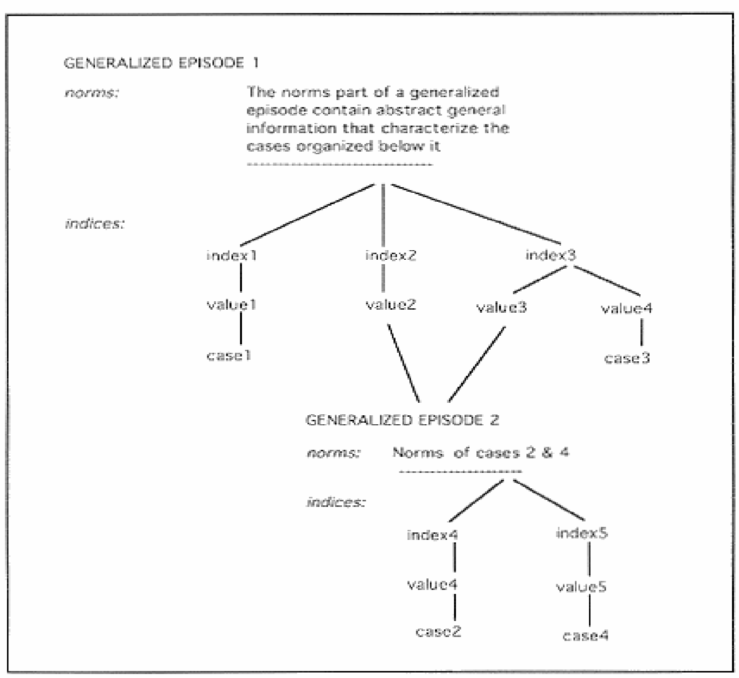
Kolodner (1993) offers a set of possible preference heuristics to help organize the storage and retrieval of cases. These include:

1. *Goal-directed preference.* Organize cases, at least in part, by goal descriptions. Retrieve cases that have the same goal as the current situation.
2. *Salient-feature preference.* Prefer cases that match the most important features or those matching the largest number of important features.
3. *Specify preference.* Look for as exact as possible matches of features before considering more general matches.
4. *Frequency preference.* Check first the most frequently matched cases.
5. *Recency preference.* Prefer cases used most recently.
6. *Ease of adaptation preference.* Use first cases most easily adapted to the current situation.

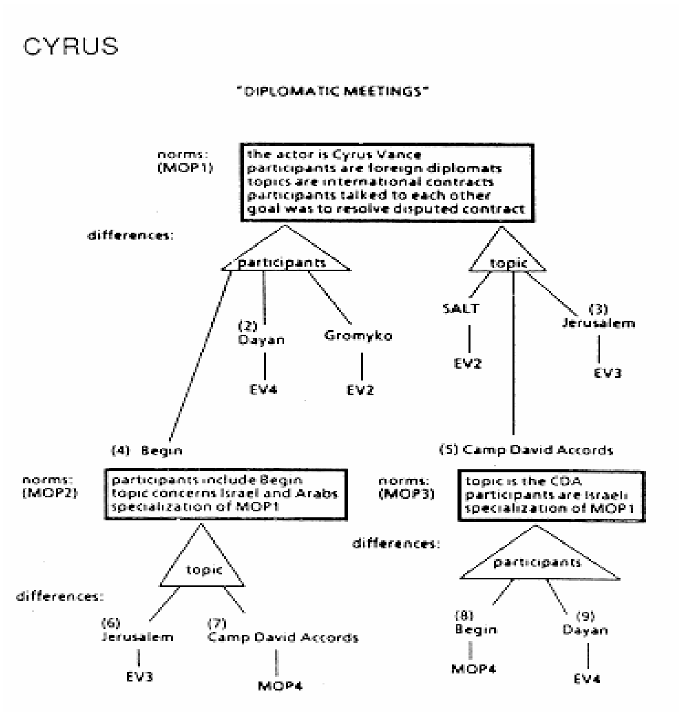
Memory organization

- Case representation formalism
  - attribute-value sets
    PROTOS, CASEY
  - structured representations
    CHEF
- Flat (or almost flat) index structures
  - feature-case (or via category)
    PROTOS
- Hierarchical index structures
  - dynamic episodic memory
    CYRUS, CASEY

Dynamic Memory
(Scank & Kolodner 83)



Example



The ”knowledge-intensiveness” scale of CBR

- **No explicit gen. knowledge**
- **A lot of cases**
- **A case is a data record**
- **Simple case structures**
- **Global similarity *metric***
- **No adaptation**
- **Learning is simple storage**

- **Substantial gen. knowledge**
- **Not very many cases**
- **A case is a user experience**
- **Complex case structures**
- **Sim. assessm. is an *explanation***
- **Knowledge-based aptation**
- **Knowledge-based learning**

CBR methods
The Data-- Knowledge Dimension

- ♦ Data intensive - Knowledge poor
  - A case is a data record
  - Similarity asessment based on simple metric

- ♦ Knowledge intensive - Data Poor
  - A case is a user experience
  - Similarity asessment is an explanation process

- ♦ Both knowledge and data intensive
  - Multiple case contents
  - Multiple similarity asessment methods

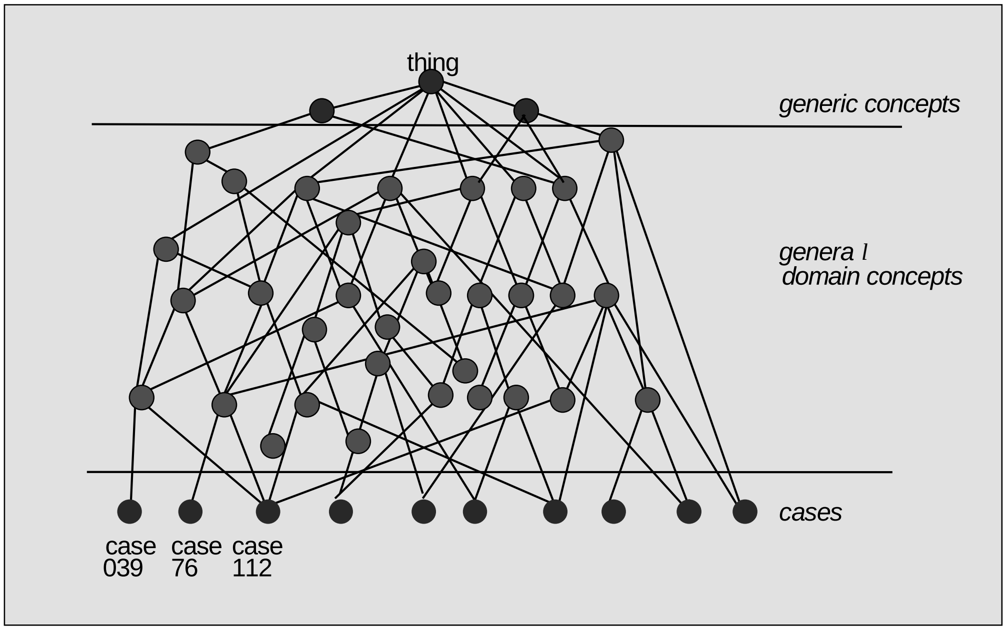
CREEK

- Case-based reasoning in open and weak theory domains; diagnosis problems (appl.: oil-well drilling, medicine)

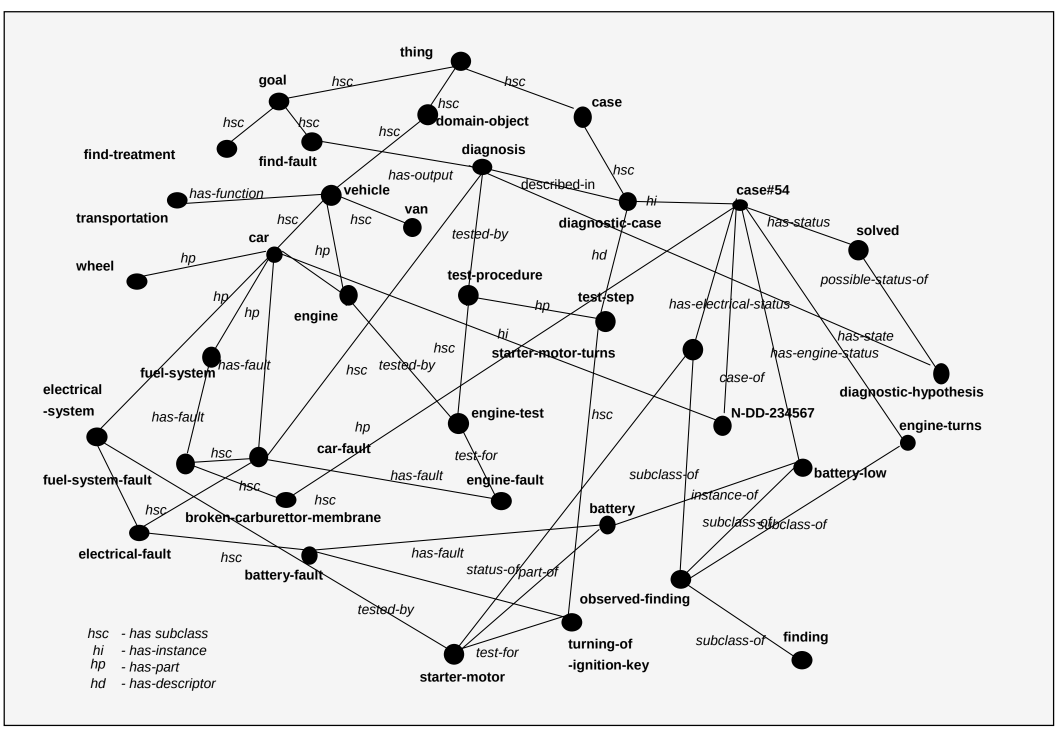
- Problem description is problem solving goal, solution constraints, and list of findings
  Solution is (one or more) diagnoses and repairs

- Knowledge types are
  - case memory of findings to solutions, indexed by relevant findings; cross-case indexes to neighbouring cases and between diagnosis and treatments
  - general domain knowledge as deep relationships or heuristiv rules
  - all knowledge integrated into a single semantic network of concepts and relations
  - each concept and each relation explicitly represented as frames

CreekL Knowledge Types



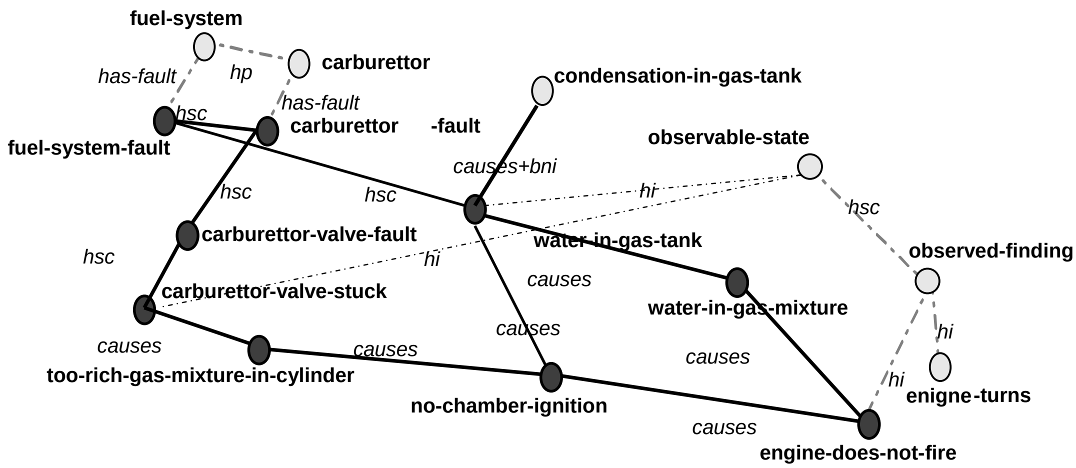
Tangled CreekL Network



case#54

instance-of	value	car-starting-case diagnostic-case
has-task	value	find-car-starting-fault
has-status	value	solved
of-car	value	N-DD-234567
has-fault	value	carburettor-valve-stuck
has-fault-explanation	value	carburettor-valve-stuck causes too-rich-gas-mixture-in-sylinder causes no-chamber-ignition causes engine-does-not-fire
has-repair	value	replace-carburettor-membrane
has-electrical-status	value	battery-low starter-motor-turns
has-engine-status	value	engine-turns engine-does-not fire
has-ignition-status	value	spark-plugs-ok
has-weather-condition	value	low-temperature sunny
has-driving-history	value	hard-driving

Explanation Structure



hsc = has-subclass
hi = has-instance

CREEK

- **Retrieve**
  - context focusing by spreading activation in the semantic netowrk, followed by
  - index retrieval of possible cases, followed by
  - explanation-driven selection of best match
- **Reuse**
  - attempts to copy solution from matched case
  - explanation-driven adaptation, by combining explanantion of retrieved case with general domain model
- **Revise**
  - user evaluates and gives feedback
  - case status info kept and used in case selection and reuse
- **Retain**
  - attempts to merge the two cases
  - stores relevant findings, sucessful and failed solutions, and their explanations
  - updating the strength of indexes

CBR systems development

- Two basic approaches:
  - bottom-up from data
  - top-down knowledge modeling

  How to combine the two is the big issue.

- For a particular application, a breakdown of knowledge and information into case-specific and general is needed.

  There has to be a number of cases available.

- Knowledge acquisition problem is in general still hard.

  KA methodologies needs to incorporate the 'case view'.

Help Desk Applications

- General help and advice, fault finding, maintenance, manual browsing, ...
- Primary CBR application type so far
- Facilitates the *retrieval* of similar past cases, and leaves the *reuse* of cases to the user
- Data and information get grouped according to the problem situations where they occurred.
- Market potential due to service costs, complexity of equipment, job instability, training of personell, ...
- Learing ability in CBR enables capturing of new experience as a 'rutine operation'.

Potential problems

- Capturing expertise is difficult. CBR helps solving some problems but also introduces some.
- Building case bases from exisiting data bases is difficult. Data mining methods may help.
- Methods for sustained learning are not welll developed yet.
- Many cases are often needed for sufficient coverage of domain. General knowledge may help here.
- Development tools are only 1. generation

A stepwise approach

- Start by viewing cases as information, i.e. to be interpreted and reasoned with by the user.

  This enables information that normally is scattered and fragmented to be retrieved on the basis of previous situations where it was created or used.

- Once the manual reuse of cases has been tested, additional reasoning and learning capabilities should be added.

Some applications

- CLAVIER (Lockheed)
  - Autoclave loading
- CaseLine (British Airways)
  - Aircraft maintenance and fault finding
- PRISM (Chase Manhattan Bank)
  - Telex classifier and router
- 'Valve assistant' (General Dynamics)
  - Pipeline valve selection
- SMART (Compaq)
  - Compaq products diagnosis
- SQUAD (NEC Corp)
  - Management of SW quality control knowledge
- QDES (Nippon Steel)
  - Design reuse

Some commercial tools

KATE-CBR	(Acknosoft)
ART-Enterprise	(Brightware)
ESTEEM	(Esteem Software Inc.)
Easy Reasoner	(Haley Enterprise)
CasePower	(Inductive Solutions)
ReMind	(Intelligent Appl. /Cognitive Systems)
CasePoint	(Inference)
ReCall	(ISoft)
CBR-Works	(TechInno)
...	

The advantages of case-based reasoning include:

1. The ability to encode historical knowledge directly. In many domains, cases can be obtained from existing case histories, repair logs, or other sources, eliminating the need for intensive knowledge acquisition with a human expert.
2. Allows shortcuts in reasoning. If an appropriate case can be found, new problems can often be solved in much less time than it would take to generate a solution from rules or models.
3. It allows a system to avoid past errors and exploit past successes. CBR provides a model of learning that is both theoretically interesting and practical enough to apply to complex problems.
4. Extensive analysis of domain knowledge is not required. Unlike a rule-based system, where the knowledge engineer must anticipate rule interactions, CBR allows a simple additive model for knowledge acquisition. This requires an appropriate representation for cases, a useful retrieval index, and a case adaptation strategy.
5. Appropriate indexing strategies add insight and problem-solving power. The ability to distinguish differences in target problems and select an appropriate case is an important source of a case-based reasoner's power; often, indexing algorithms can provide this functionality automatically.

The disadvantages of case-based reasoning include:

1. Cases do not often include deeper knowledge of the domain. This handicaps explanation facilities, and in many situations it allows the possibility that cases may be misapplied, leading to wrong or poor quality advice.
2. A large case base can suffer problems from store/compute trade-offs.
3. It is difficult to determine good criteria for indexing and matching cases. Currently, retrieval vocabularies and similarity matching algorithms must be carefully hand crafted; this can offset many of the advantages CBR offers for knowledge acquisition.

The combination of rule-based and case-based systems can:

1. Offer a natural first check against known cases before undertaking rule-based reasoning and the associated search costs.
2. Provide a record of examples and exceptions to solutions through retentioin in the case base.
3. Record search-based results as cases for future use. By saving appropriate cases, a reasoner can aviod duplicating costly search.

The combination of model-based and case-based systems can:

1. Give more mature explanations to the situations recorded in cases.
2. Offer a natural first check against stored cases before beginning the more extensive search required by model-based reasoning.
3. Provide a record of examples and exceptions in a case base that can be used to guide model-based inference.
4. Record results of model-based inference for future use.