


The University of Auckland

CS.760

Case-Based Reasoning 4
Dr. Ian Watson

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz




The University of Auckland

Applications & Tools

- Discuss some influential CBR applications
- Then have a bit of fun today
 - Look at CBR-Works
 - Look at AIAI CBR Tool

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz



The University of Auckland

Problem classification

<ul style="list-style-type: none"> ■ classification tasks <ul style="list-style-type: none"> ■ diagnosis - what type of fault is this? ■ prediction - what happened when we saw this pattern before? ■ estimation ■ planning ■ good for CBR 	<ul style="list-style-type: none"> ■ synthesis tasks <ul style="list-style-type: none"> ■ engineering design ■ planning ■ scheduling ■ harder for CBR
--	---

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

The University of Auckland

Problem classification

```

graph TD
    CBR[CBR Systems] --> Classification[Classification Tasks]
    CBR --> Synthesis[Synthesis Tasks]
    Classification --> Prediction
    Classification --> Assessment
    Classification --> Diagnosis
    Classification --> Planning
    Diagnosis --> Medical[Medical Diagnosis]
    Diagnosis --> Equipment[Equipment Failure]
    Synthesis --> Design
    Synthesis --> Planning
    Synthesis --> Configuration
  
```

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

The University of Auckland

Task-method decomposition

```

graph TD
    CBR[CBR] --> retrieve
    CBR --> reuse
    CBR --> revise
    CBR --> retain
  
```

we're going to look at each activity in turn

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

The University of Auckland

Task-method decomposition

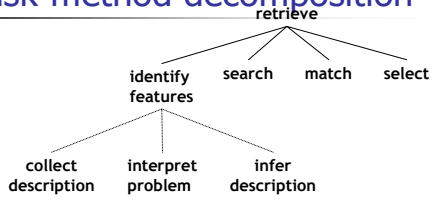
```

graph TD
    retrieve --> identify[identify features]
    retrieve --> search
    retrieve --> match
    retrieve --> select
  
```

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz



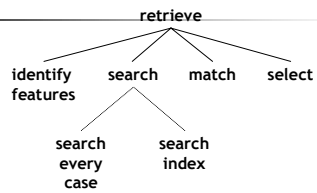
Task-method decomposition



© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz



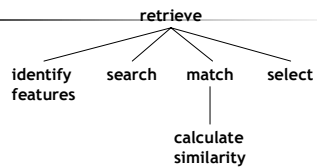
Task-method decomposition




© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz



Task-method decomposition




© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

 **Task-method decomposition**

```

    graph TD
      retrieve --> identify_features[identify features]
      retrieve --> search
      retrieve --> match
      retrieve --> select
      select --> order_cases[order cases]
      select --> user_selects[user selects cases]
      select --> selection_criteria[selection criteria]
  
```


© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

 **Task-method decomposition**

```

    graph TD
      CBR --> retrieve
      CBR --> reuse
      CBR --> revise
      CBR --> retain
      reuse --> use_solution[use solution]
      reuse --> use_solution_method[use solution method]
  
```


© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

 **Task-method decomposition**

```

    graph TD
      CBR --> retrieve
      CBR --> reuse
      CBR --> revise
      CBR --> retain
      revise --> evaluate_solution[evaluate solution]
      revise --> repair_faults[repair faults]
  
```


© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

 **Task-method decomposition**

```

    graph TD
      revise --> evaluate_solution[evaluate solution]
      revise --> repair_faults[repair faults]
      evaluate_solution --> evaluate_in_vivo[evaluate in vivo]
      evaluate_solution --> evaluate_in_model[evaluate in model]
      repair_faults --> identify_faults[identify faults]
      repair_faults --> self_repair[self repair]
      repair_faults --> user_repair[user repair]
  
```


© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

 **Task-method decomposition**

```

    graph TD
      CBR --> retrieve
      CBR --> reuse
      CBR --> revise
      CBR --> retain
      retain --> add_case[add case]
      retain --> integrate_case[integrate case]
  
```


© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

 **Task-method decomposition**


```

    graph TD
      retain --> add_case[add case]
      retain --> integrate_case[integrate case]
      add_case --> update_case_base[update case-base]
      add_case --> re_index[re-index]
      integrate_case --> adjust_indexes[adjust indexes]
      integrate_case --> adjust_weights[adjust weights]
  
```

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz




Case study - Lockheed




- PROBLEM - how to optimise the loading of an autoclave for curing composite materials
- different materials need different heating & cooling procedures
- materials interact with each other in the autoclave
- mistakes are **VERY** costly

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz




Case-study - Lockheed




- 2 experienced operators relied on plans of previously successful layouts
- New layouts were adapted from old
- If successful they were added to a library
- they wanted to develop a decision support tool to assist experts and to retain expertise as a corporate asset

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz





Case-study - Lockheed




- Lockheed had **NO** model of the autoclave
- the manufacturers could not provide one
- layouts do not repeat much
- materials are constantly changing
- designs constantly change
- elements interact

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

 **Case study - Lockheed** 

- their system was implemented in 1990
- CLAVIER started with 20 successful layouts
- CLAVIER now has hundreds of successful layouts
- it retrieves a successful layout or adapts one 90% of the time
- acts as a corporate memory

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

 **Further reading**

- Watson, I. (1997). *Applying Case-Based Reasoning: techniques for enterprise systems*. Morgan Kaufmann Inc. San Francisco Calif. US
- Bergman, R. & Althoff, K.-D. (1998) *Methodology for Building CBR Applications*. In, *CBR Technology: from foundations to applications*. Lenz, M. et al. (Eds.) Springer Verlag, LNAI #1400 pp.299-326

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz
