Case-Based Reasoning (Lab)

10/28/02
CASUEL: Common Case Representation Language

- CASUEL is a syntax for representing the data structures, types and data needed for building a CBR application ([http://wwwagr.informatik.uni-kl.de/~bergmann/casuel/](http://wwwagr.informatik.uni-kl.de/~bergmann/casuel/))
  - The philosophy of the language is object-oriented, but CASUEL is not intended to be a computer implementation of an object-oriented language
  - It is merely a syntax for describing all the information that is relevant for an application
Using CASUEL, the following are declared and defined:

- Basic objects of the application domain. e.g., a patient in a medical application.
- Relations between the basic objects. e.g., the component subparts of an object.
- Slots used to describe the basic objects. Example: the age, sex and medical history of a patient.
- Type of slots that specify:
  - Whether a slot will accept a number, like "age", a nominal value, like "sex"
  - The range of allowed values for the slot. "Age" must be between 0 and 120
    "sex" is either "male" or "female"
  - The number of allowed values. "Age" can only take a single value but "medical history" can have a conjunction of several values.

Once this general descriptive model has been designed and formalized in CASUEL, the cases are also represented in CASUEL according to this model.

- e.g., a particular patient named Smith is 36 years old, male and with a history of allergies to penicillin and an operation to his right kidney.

A case is a collection of objects

- It is defined by the CASUEL statement defcase. The objects in a case are linked with one-another by relations.
CABATA (CAse BAseB Travel Agency)

• CABATA chooses one of its stored cases describing the past holiday trips as a suggestion for a new trip satisfying user-specified conditions

• \~wjpai\k/cbr/travel-domain
  – reise.domain
  – reise.objects
  – reise.slots
  – reise.types
  – reise.cases
Another objective of CATABA was to learn ‘determination rules’

- How certain attribute values influence other attributes of the case or diagnosis

Type #1: dynamic similarity assessment may be influenced by increasing or decreasing the importance of particular features if certain conditions hold

  e.g., IF HOLIDAY_TYPE = CITY THEN SEASON IS OF LESS INTEREST

Type #2: rules serve as restrictions/constraints when demanding that a particular feature is of a certain value

  e.g., IF REGION IsA SEA THEN REGION MUST NOT BELONG TO MOUNTAINS
HICAP: Hierarchical Interactive Case-based Architecture for Planning

• A multimodal, mixed-initiative, and domain-independent plan authoring (i.e., elicitation) tool

• Planning a course of action is difficult, especially for large hierarchical organizations (e.g., the U.S. Navy) that constrain plans with guidelines (e.g., doctrine) and assign resources, both material and human, to tasks
  – A concrete plan must adhere to guidelines but should also exploit an organization's experiential knowledge where appropriate (e.g., standard procedures for solving tasks, previous experiences when reacting to unanticipated situations)
  – Case-based reasoning (CBR) can be used to capture and share this type of knowledge, which can potentially reduce the time required to generate a plan and also increase plan quality
HICAP (Cont’d)

• Intelligent plan formulation tool with the following characteristics:
  – *Guidelines-driven*: Use an organization's guidelines to constrain plan generation
  – *Interactive*: Allow users to control the plan authoring process, and edit any detail of the plan
  – *Provide Case Access*: Index plan segments from previous problem-solving experiences, and retrieve them for users to incorporate into a new plan, if warranted by the current planning scenario
  – *Perform Bookkeeping*: Maintain information on the status of and relations between task responsibilities and individuals in the organization's hierarchy
  – *Partially automated*: Automatic plan generation may be used for portions of the plan as desired
HICAP (Cont’d)


1. HTE handles planning problems using hierarchical plan generation
2. The case-based planning component of HICAP, Nacodae/HTN, typically performs three steps: retrieval, revision, and retention.
   1) During a conversation, cases are ranked according to the proportion of their question,answer pairs that match the current state.
   2) In addition, the user can revise their answers to previously selected questions, which can modify case rankings.
   3) It evaluates whether any case "subsumes" another case (i.e., whether its question,answer pairs are a proper subset of the question-answer pairs of another case).
3. Without a complete domain theory, HICAP cannot guarantee it will produce a correct plan for all possible states. But obtaining a complete domain theory is often difficult, if not impossible. Lessons can help fill gaps in a domain theory so that, when reused appropriately during planning, they can improve plan performance. This is the motivation for applying lessons while using HICAP.
SIROCCO: System for Intelligent Retrieval of Operationalized Cases and COodes

- A system for retrieving principles and past cases. This paper presents empirical evidence that the operationalization information contained in extensionally-defined principles can indeed be leveraged to predict relevant principles and past cases
  - General principles are abstract rules intended to guide decision makers in making normative judgments in domains like law, politics, and ethics
  - It is difficult, however to define principles in an intensional manner so that they may be applied deductively
  - The problem is the gap between abstract, open-textured principles and concrete facts.
  - When expert decision makers rationalize their conclusions in specific cases, they often link principles to the specific facts of the cases
    - These expert-defined associations between principles and facts provide extensional definitions of the principles
    - The experts operationalize the general principles by linking them to the facts.
SIROCCO (Cont’d)

• An interpretive case-based reasoning (CBR) program
  Interpretive CBR is a sub-field of case-based reasoning in which complex, ill-structured, and highly linguistic fact situations are evaluated in the context of previous experience
  • It can retrieve cases over a wider range of factual scenarios
  • It does not make arguments for or against a conclusion
  • It provides suggestions that can help a human construct a reasoned argument.

It operates by
1) Accepting a target case expressed in the Ethics Transcription Language (ETL),
2) Searching for relevant information in a case base of source cases expressed in the Extended Ethics Transcription Language (EETL), and
3) Producing suggested code provisions and past cases, as well as other suggestions.
SIROCCO (Cont’d)

http://sirocco.lrdc.pitt.edu/sirocco/servlet/Sirocco Weblink
Real-life Application

http://www.egain.com/egainassistant/