Model-Based Reasoning About Cases

Juho Rousu and Robert J. Aarts[1]

VTT Biotechnology and Food Research
P.O. Box 1500, FIN-02044 VTT, Finland
Juho.Rousu@vtt.fi

Abstract

Our bioprocess recipe planner Sophist (Aarts & Rousu, 1996) uses a qualitative model to represent a large amount of domain knowledge. This knowledge is used to analyse differences between cases, to construct adaptations, to evaluate suggested plan changes and to explain outcomes. The module for Qualitative Modeling in a CBR system (QMC) was especially developed to support these Case-Based Reasoning tasks. Hence, QMC is tightly integrated in the Sophist framework. Both Sophist (Aarts & Rousu, 1996, Rousu & Aarts, 1996) as well as QMC (Aarts & Rousu, 1997) have been described earlier so this paper briefly describes planning in Sophist and then focuses on some interesting issues that arose during development and deployment of this multimodal reasoner.

Planning in Sophist

Sophist was designed to support planning in *continuous* domains, that is the planning problems are given in terms of continuous quantities. The primary mode of reasoning in Sophist is CBR. When the system is asked to construct a plan for a new problem it retrieves a template plan that was successfully used to solve a similar problem. The differences between the new and earlier problem are analyzed and the planner *adapts* the template plan. When the new plan has been executed the outcome of the case is analyzed and the case is added to the casebase. In Sophist a qualitative model of the domain is used to support each of these CBR tasks.

Planning in a very simple flight domain is used as an example (see Aarts & Rousu, 1996, for an example of a more realistic application). In this domain planes have to fly a required distance with a particular payload and a given amount of fuel. A single flight is a case with a given plane, payload and amount of fuel. The task of the case-based reasoner is to construct a flight path such that the requested distance is flown. A simple qualitative model of this domain was constructed (see Figure 1) and a handful of cases were conceived. These cases have flight paths consisting of three flight segments: an ascend, a period of level flight and a descend.

The model is a graph of process, variable and rule *nodes* linked by *influences*. It is a kind of spreading activation network, with different interpretations of activity for different types of nodes. For a rule the intuitive interpretation is "truth", for a process it is "rate", and for a variable it is "trend". These concepts support the construction of rather comprehensive models in a variety of domains. Note that, although the links have associated weights, the models are primarily qualitative! Without cases, it is impossible to infer accurate quantitative information from such a model, for instance it is impossible to determine even a rough quantitative flight plan from the model presented in Figure 1.

In Sophist reasoning about cases always starts by mapping of case features to the appropriate nodes in the model (nodes that correspond to case features are indicated by **bold** boxes in Figure 1). In general, the case-based reasoner maps some features of a case to nodes in the graph, the influences are traversed and then relevant nodes return appropriate objects. Dependent on the task, particular features are mapped and different nodes should be traced. This is best illustrated with an example of plan construction. The example is based on the simple flight model shown in Figure 1. The task is to construct a plan for Flight 6 from the template case Flight 4 (see Table 1).

First, the case-based reasoner constructs *case differences*. Case differences are structures that hold the values of both a template and a target case for a particular feature. In the flight example, one of the constructed case differences is "a greater requirement for distance". These case differences are then mapped to an appropriate model node. For example, "a greater requirement for distance" maps to the `distance` variable in the model because the subject of the difference is a *minimum* (a kind of constraint) for the *quantity* named "distance". That quantity is shared with the model variable `distance`. (Figure 1). Actual analysis of the difference starts by traversing the influence graph to processes that in principle could compensate for the difference. Such processes return an *adaptation goal*. An adaptation goal has an associated importance value that, in turn, is computed by the measurement nodes from the

[1] Current address: Nokia Telecommunications, P.O. Box 370, FIN-00045 NOKIA GROUP, robert.aarts@ntc.nokia.com

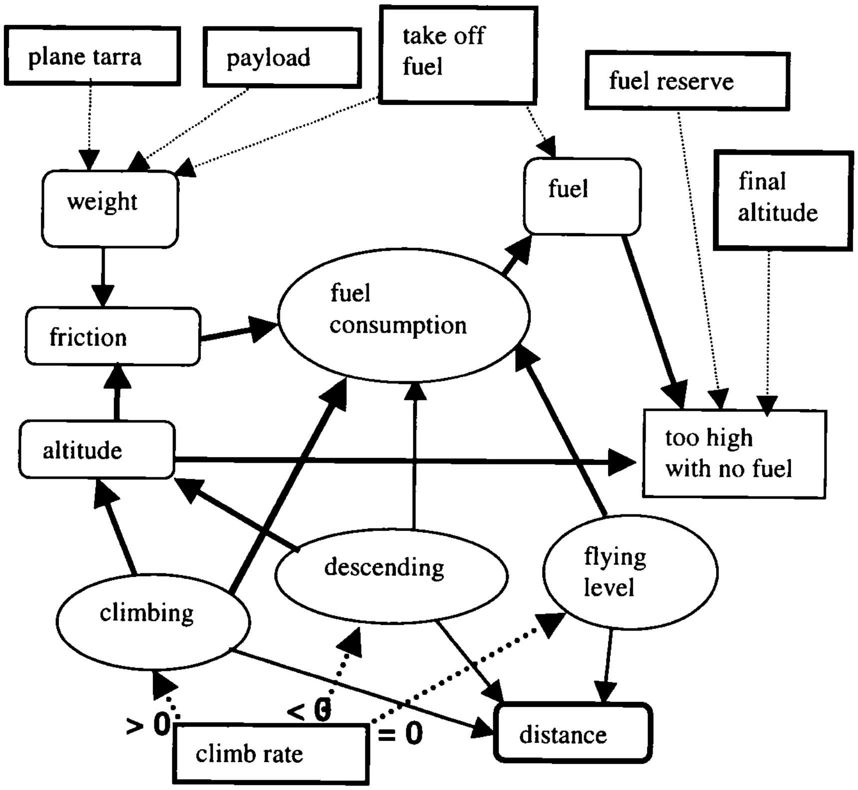

Figure 1. The influence graph of the simple flight domain. Ellipses denote *processes*, rounded rectangles are *variables* and rectangles indicate *primitives* and *rules*. Straight arrows are *influences*; dashed arrows are *support* links. Arrow print weights indicate the relative strength of link. **Bold** rectangles indicate nodes that are mapped onto case features.

case differences, using simple heuristics or predefined fuzzy sets. According to the example model, a greater distance can be achieved by an increase in `climbing`, `flying level` or `descending`, as each of these processes influence `distance`.

Next, the model is asked for adaptations of the template plan for each of the adaptation goals. The model maps the adaptation goal to one or more appropriate nodes (as for case differences) and asks that node for relevant adaptations. Variables pass this request on along their influences, but processes also ask their conditions for adaptations. This recursion bottoms out at support links from measurement nodes to other nodes. The measurement checks if any of the template operations activates the link. Such an operation is then asked for an adaptation that meets the adaptation goal. For example, the adaptation returned for "an increase of flying" is "an increase in the length of the level flight segment" of Flight 4. Similarly, it possible to check for side-effects by propagating the proposed adaptation through the model. Finally the adaptations are ranked by *expected benefit*, and the most effective adaptations are made.

The size of an adaptation is calculated on the basis of the importance of the adaptation goals and the weights of the relevant influences. In the flight example, the planner decides to increase the time of level flight by 1 minute. From the outcome one can see that the adaptation was in the right direction but too small. However, a new case, closer to solving the original problem, can now be added to the casebase.

Discussion

Obviously a qualitative model is a rich source of problem solving power. The availability of a model, even an incomplete QMC model, enhances the problem solving power of a case-based reasoner when the casebase is small. This as opposed to applications where cases are used to enhance the performance of a (complete) model-based reasoner (Koton, 1989; Portinale & Torasso, 1995).

When a casebase grows, problem solving becomes more accurate, as suitable template cases are found more frequently. Note, however, that finding a good template in a large casebase can be computationally expensive (Smyth & Cunningham, 1996). Moreover, in a complex domain the interesting problems are often those that are somewhat abnormal, and that are not encountered frequently. Having a model to point out the correct direction of change does not guarantee that the outcome will be acceptable, but it certainly helps to make progress towards the goal, nevertheless.

It would be very helpful if the appropriate *size* of an

Table 1. A template case, Flight 4, and a target case, Flight 6. The task of the planner is to construct a flight path for Flight 6 on the basis of the flight path of Flight 4. The suggested plan and simulated outcome for Flight 6 are in *italic*. (from Aarts & Rousu, 1997).

	Flight 4	Flight 6
Distance to goal	500	600
Plane Payload Take-off fuel	Liner 20000 kg 15000 kg	Liner 25000 kg 10000 kg
Flight path	Ascend to 10000 in 20′ at 400 Fly at 10000 for 20′ at 700 Descend to 0 in 15′ at 400	*Ascend to 10000 in 20′ at 400* *Fly at 10000 for 21′ at 700* *Descend to 0 in 17′ at 400*
Distance flown Fuel consumed Final altitude	500 km 5743 kg 0 m	*533 km* *6084 kg* *0 m*

adaptation could be computed. This would require a model with much better, but hard to obtain, quantitative knowledge. Alternatively, the size of an adaptation could be determined on the basis of the effectiveness of previous, similar, adaptation episodes. This is a form of case-based adaptation (Leake *et al.*, 1995).

QMC models work as a universal knowledge source that can be utilised for retrieval, adaptation, evaluation, and explanation of outcomes alike. In this way, QMC is somewhat orthogonal to the idea of four knowledge containers by Richter (1995): we emphasize the importance of unifying those four *types* of knowledge as much as possible. For example, embedding some domain knowledge in the distance metric hides it from an adaptation algorithm. During adaptation one would like to utilise any information that is available about to the importance of differences and pitfalls.

Naturally model construction is not trivial, although a qualitative model is often far easier to construct than a quantitative (mathematical) model or a set of adaptation rules. A crucial question is what information should be represented by the cases and what by the model. Several arguments suggest to represent quantitative information in cases. Quantitative aspects are likely to be very context-specific and are easily represented as case features, whereas identification of quantitative models is often impossible. On the other hand qualitative aspects are often better represented in a model. For example, it is space consuming to have a trajectory of `fuel consumption` as a case feature, and difficult to use such a feature. Also, the more qualitative the knowledge is, the more context-independent it seems to be. For example, the law of thermodynamics that depicts that energy flow is always from warm to cold is literally universal, but the exact amount of energy transferred in a heat exchanger can be very difficult to calculate.

Explanation based reasoning as in SWALE (Schank & Leake, 1989) hints at possibilities to *learn* the model with the aid of expectation violations. The model could be asked to construct explanations for the observed outcome of the case, given the plan and the situation, that is. If the model can indeed explain all the observations, the model is valid. If however, the model cannot explain an observation, it obviously needs to be revised. Work by DeJong (1994) demonstrates how model learning might be achieved in continuous domains. Other research on model identification, i.e. determining the underlying "deep" variables from system behaviour, includes work by Say & Kuru (1996) and Nayak & Joskowicz (1996).

References

Aarts, R.J. & Rousu, J. 1996. Towards CBR for bioprocess planning. In Smith I., Faltings, B., (Eds.): Proceedings of EWCBR-96, Lausanne, *Lecture Notes in Artificial Intelligence*, **1186**: 16-27.

Aarts, R.J. & Rousu, J. 1997. Qualitative Knowledge to Support Reasoning About Cases. In Leake, D. B., Plaza, E., (Eds.): Proceedings of ICCBR-97, Providence, *Lecture Notes in Artificial Intelligence*, **1266**: 489-498.

DeJong, G. F. 1994. Learning to plan in continuous domains. *Artificial Intelligence* **65**: 71–141.

Koton, P. 1989. *Using experience in learning and problem solving*. Massachusetts Institute of Technology, Laboratory of Computer Science (Ph.D. diss., October 1988), MIT/LCS/TR-441.

Leake, D.B., Kinley, A. & Wilson, D. 1995. Learning to Improve Case Adaptation by Introspective Reasoning and CBR. In: Veloso, M. & Aamodt, A. (Eds.): Proceedings ICCBR-95, Sesimbra, *Lecture Notes in Artificial Intelligence*, **1010**: 229–240.

Nayak P. & Joskowicz, L. 1996. Efficient compositional modeling for generating causal explanations. *Artificial Intelligence* **83**: 193-227.

Portinale, L. & Torasso, P. 1995. ADAPtER: An Integrated Diagnostic System Combining Case-Based and Abductive Reasoning. In: Veloso, M. & Aamodt, A. (Eds.): Proceedings ICCBR-95, Sesimbra, *Lecture Notes in Artificial Intelligence*, **1010**: 277-288.

Richter, M. 1995. The similarity Issue in CBR : The knowledge contained in similarity measures, Invited talk at ICCBR -95, Sesimbra.

Rousu, J. & Aarts, R.J. 1996. Adaptation Cost as a Criterion for Solution Evaluation. In Smith I., Faltings, B., (Eds.): Proceedings of EWCBR-96, Lausanne, *Lecture Notes in Artificial Intelligence*, **1186**: 354-361.

Say, A.C.C. & Kuru, S. 1996. Qualitative system identification: deriving structure from behavior. *Artificial Intelligence* **83**: 75-141.

Schank, R.C. & Leake, D.B. 1989. Creativity and Learning in a Case-Based Explainer. *Artificial Intelligence* **40**: 353-385.

Smyth, B. & Cunningham, P. 1996. The Utility Problem Analysed. In Smith I., Faltings, B., (Eds.): Proceedings of EWCBR-96, Lausanne, *Lecture Notes in Artificial Intelligence*, **1186**: 392-399.