THE ELSA AVALANCHE PATH ANALYSIS SYSTEM: AN EXPERIMENT WITH REASON MAINTENANCE AND OBJECT-BASED REPRESENTATION

EXTENDED ABSTRACT

Laurent Buisson and Jérôme Euzenat

Division Nivologie
CEMAGREF
BP 76
F-38402 SAINT-MARTIN D'HÈRES
laurent@beaufort.grenet.fr

ILOG,
2 avenue Galliéni,
BP 85
F-94253 GENTILLY Cedex
euzenat@ilog.fr

ELSA is an application concerning avalanche path analysis which takes advantages of a reason maintenance system. In order to fully describe it, the tool on which the ELSA application is developed — Shirka/TMS — is first described. It is noteworthy that the RMS on Shirka is a special one. It is only used for cache consistency maintenance. As a consequence, the importance of Shirka/TMS in ELSA is in preserving cache consistency rather than defaults assumptions and backtracking. Then, the processing of the ELSA system is presented, emphasizing on the use of the RMS: the RMS of Shirka is critical for the performances of the whole system. This is illustrated in the third part in which are given some comparison of the use of ELSA with and without its RMS, in order to highlight the advantages of such a device.

SHIRKA/TMS: OBJECT-BASED REPRESENTATION AND RMS

Shirka is a traditional object-based knowledge representation system written in Lisp [9]. Everything, in Shirka, is an object (including inference methods). Each object belongs to a class which defines its structure — in terms of a list of fields and constraints on the field values — and its inferential capabilities — in terms of inference methods used in order to determine the values of unfilled fields. Inference methods are among value passing, procedural attachment, pattern matching and default values.

Classes are organized in a direct acyclic graph structured by the *a-kind-of* relationship between classes. This relationship enables inheritance from a class to its specializations. Inheritance is used through class refinement — a class strongly inherits, i.e. possesses, its constraints on fields from its super-class — and inference specialization — a class weakly inherits, i.e. inherits by default, its inference methods.

A RMS has been implemented on Shirka [7, 6]. It is a classical "justification-based"-TMS except that it records and propagates field values. It takes into account the fact that, in an object-based knowledge representation such as Shirka, applicable inference methods for an object are ordered. Thus, the propagation procedure of the system is very simple, since it is deterministic. Moreover, Shirka has its own consistency recovery mechanism: it is able to detect (local) inconsistencies and simply declares the inference that generates it as failing. Shirka/TMS takes advantage of these capabilities in order to avoid backtracking. Since Shirka is not provided with global inconsistencies (constraints), it is a well-defined deterministic inference system.

The RMS records the successive tentatives to infer a field value in an ordered structure and each tentative (or effective) inference is linked to its premises like with a justification. So, the currently valid value is cached away. If some part of the currently valid inference become invalid, the current value will be cancelled and the cancelling propagated through the dependency network. If something changes in the knowledge base that can enable, an inference method that had previously failed, to fire the system will be able to cancel the current value and the re-infering will take place with the new available method in case of a new query on the field. As a consequence, the RMS of Shirka/TMS is very simple and efficient.

The underlying assumption of the implementation of a RMS in an object-based knowledge representation is that the base is queried very often (w.r.t. modifications). The performances are very attractive because re-infering is avoided (and thus, the answers are given very quickly). On another hand, the modifications — that are safely dealt with — and initial inferences are processed more slowly. This assumption was enforced by the observations made with benchmark tests and, moreover, with the ELSA application.

ELSA: AN AVALANCHE PATH ANALYSIS SYSTEM

ELSA [1, 2] is a problem solving environment which offers to a snow specialist the different tools available in order to perform an avalanche path analysis and choose the best protection devices. It has been tested and tuned on four real size terrains of the french alps and is now used by CEMAGREF specialists. ELSA is a knowledge-based system which uses both symbolic simulation based on expert knowledge and numerical simulation based on fluid mechanics conservative laws.

Two phases can be distinguished in the use of ELSA: the first phase (terrain inference) builds a terrain representation relevant to the specialist point of view while the second one (scenario simulation) simulates a meteorological scenario until an avalanche fires. Both phases take advantage of the caching of inference results and of the revision aspect of the RMS in order to find out what must change or remain after some modification in the data. In this paper the emphasis is on terrain inference.

Because of the spatial extension of the phenomena involved in snow avalanches (snow-drift, snow-cover stability, fracture propagation, avalanche flowing...), ELSA needs spatial information on the path. In order to get this information or to use it, ELSA performs an actual spatial reasoning as it has been defined in [2]. In fact, from poorly relevant spatial knowledge such as contour line, vegetation or ridge maps, ELSA must infer the definition of special units of terrain, called "small panels" by snow specialists, and which are relevant for analysis (they are homogeneous from the analysis criteria points of view). Meanwhile this definition in small panels is not sufficient and ELSA must also infer the properties of these small panels for performing its analysis. For example, here is the spatial definition of a small-panel called `pp1` (in Shirka's frame like syntax). At its creation, this small panel is defined only by the list of triangles included in it.

```
{pp1
        is-a                = small-panel ;
        contains            = tr2 tr93 }
```

In order to analysize the avalanche starting zone, ELSA requires more information and, to that extent, infers a more complete description of the small panel `pp1` (shown below). All the fields inferred by ELSA are obtained by the use of inference methods, mainly, pattern-matching inference and procedural attachment.

```
{pp1
        is-a                = small-panel ;
        area                = 6850. ;
        c-gravity           = %point-552 ;
        altitude            = 1075. ;
        diameter2D          = 115. ;
        diameter3D          = 138. ;
        slope-%             = 68.  ;
        is-in               = tende ;
        contains            = tr2 tr93 ;
        boundary-points     = po4 po6 po5 po1 ;
        connected-panels    = pp2 pp3 ;
        borders             = %border-589 %border-590 ;
        close-ridges        = ar1 ar3 ar4 ar5 ;
        above               = pp3 }
```

Reason maintenance is interesting in an interactive environment for spatial reasoning. As a matter of fact, the caching of inferences is necessary because of the size of the spatial knowledge base and the amount of inferences. In ELSA, an avalanche path can easily contain more than 500 triangles and 50 small panels and ridges. Without caching, the time taken for the inferences will forbid any interactive use of the system while ELSA is dedicated to decision support and thus needs interactive use.

But, in this kind of context, the user is also supposed to modify given knowledge. In ELSA, the user can change the vegetation of a part of a small panel (in order to simulate protection works for instance), or modify the definition of a small panel (toward a more accurate decomposition of space). As a result, the spatial properties of these small panels must be re-inferred. In order to keep the base consistent, a RMS is necessary.

Although ELSA is based on Shirka/TMS, it can take advantage of the RMS in order to manage dynamicity in spatial reasoning. Fig. 1 gives a good example of interest of such a RMS.

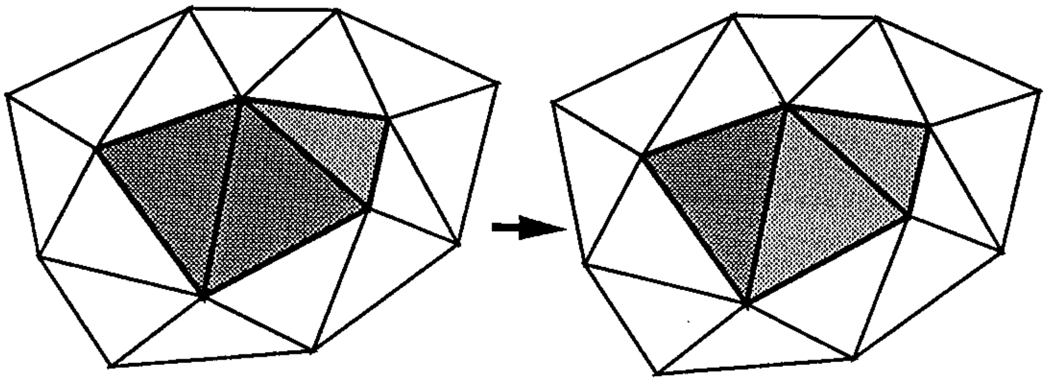


Fig. 1. In a triangulation of space, two polygons are defined through the set of triangles which are included in them. If a triangle changes its polygon, the RMS must invalidate the cached inferences concerned by these two polygons. Meanwhile, the inferences conducted on the other polygons are not modified. The invalidation remains local.

As a summary, it appears that spatial reasoning applications can take advantage of classical RMS abilities. More precisely, the spatial locality can be translated in the dependency graph.

RESULTS AND LESSONS: CONTRIBUTION OF THE RMS TO THE PERFORMANCES OF ELSA

The observation we made (comparing ELSA with or without RMS, see tab. 1 and [3] for complete results) are counter-intuitive at first sight:

1) Of course, the second call to the same inference takes no time with the RMS while, in spite of the filtering capabilities of Shirka, it still takes a while without it.
2) Even the first call is faster with the RMS than without (with a factor ≥ 2)!
3) Moreover, the time required to answer the same query against another object is reduced by a factor 8.

maintenance level	Shirka	Caching	RMS
Total 1	74.48	10.73	12.57
Total 2		81.58	97.3
Total 3 (initial inference + modification + re-inference)		175.8	134.21

Tab. 1. The first serie of tests allows to compare the advantages of caching with re-infering; the second serie is too long without caching. The two first totals concern a representative set of the inferences in ELSA. The third one concerns the total time required for infering the two first sets and re-infering them after a change such as the one of fig. 1.

So these evaluations reveal a synergistic effect between inferences. These effects can be summarized as:

Backward cone effect: there is a backward cone effect when a datum is used several times in the computation of another. This can be stated in another way: the more used the datum, the better the caching. This effect is as much interesting as the datum is expensive to compute. Backward cone effect is able to explain the results above for points (2) and (3). Intermediate performed inferences use each other several times in order to obtain the high-level (or requested) data. With the RMS, these intermediate data are computed only once. For the same reason that the inferences of different data share the same intermediate inferences, after the computation of an item, the required time to answer the same query against another object is reduced. The two former points explain why the system is also faster on the re-computation after a change.

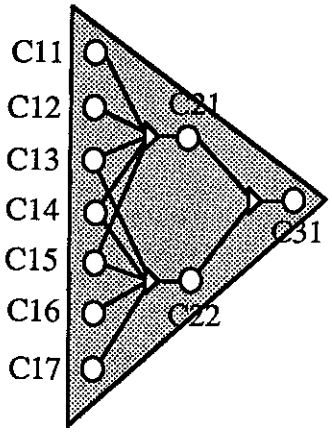
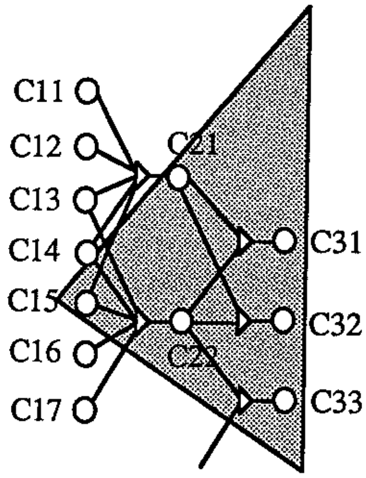


Fig. 2. In order to obtain C31, the system must infer C21 and C22 which, in turn, necessitates other inferences. Their computation can take advantage of caching because they share common inferences. This explains why the inferences produced with caching are faster even for their first computation.

Forward cone effect: the more used the datum, the worse the invalidation. As before, there is a forward cone effect when a datum is used for the inference of an important number of other pieces of knowledge. The forward cone effect is a negative effect, it reflects the necessary work in order to invalidate a cached result. It explains the classical results of observation (1) with Shirka/TMS.

Fig. 3. The whole graph represents the inferences made by the inference engine. The shaded part of the graph is invalidated after the suppression of C15. *Qualitatively*, this shaded part looks like a "forward cone". The larger is this cone, the less interesting is the RMS because the number of inferences to launch is nearer from the number of all the inferences.



Here these effects (common to RMS) are taken as external (i.e. they concern inferences) but it can be compared with the effect taken into account in RETE [4] networks by putting together the same rule patterns in order to join them only once. This allows to save time at inference time that which is paid at invalidation (retraction) time. Other works joined together a RETE algorithm and a TMS [5] and ATMS [8] respectively. They establish a justification for each pattern-joining. This allows, at a level internal to the inference, to take into account the backward cone effect (with the RETE) and enables revision (with the RMS).

A problem that we addressed [3] is: how is it possible to quantify these effects? and which conclusions to draw for the use of a RMS in a particular application. In fact, the advantage of a RMS toward rough caching is a trade-off between backward and forward cone effect. We developed a quantitative analysis of the reasoning graph in order to answer these questions. Numeric criteria defined on properties of the dependency graph are used and allow to decide if the RMS will reveal useful. It depends, of course, of other parameters such as the number of queries between two revisions.

CONCLUSION

As a conclusion, the knowledge-based system ELSA is implemented with the help of an object-based representation system provided with RMS capabilities (Shirka/TMS). While the RMS is only used in order to maintain a cache of the inferences, it has been found very useful. There is evidence that the simulation capabilities of ELSA would not have been possible without the RMS. The causes of this usefulness have been characterized in terms of forward and backward cone effects and these effects can be quantified in order to highlight their respective impacts.

The RMS in Shirka takes into account two particularities: the fact that objects store propositions in fields which do not have a boolean value and the backward chaining mode of infering field values. This could be taken into account further away by softening the lazy evaluation mode used in backward chaining and by taking into account a "replacement based"-RMS [10] able to cancel the inferences only when it is certain that they do not hold and not when some condition of their computation has changed. This should considerably reduce useless propagation (both at inference time and at invalidation time). Moreover, this is also true for the merge of RMS and constraint programming languages.

REFERENCES

[1] Laurent Buisson, "ELSA : a problem solving environment for avalanche path analysis", in: Alain Pavé, G. Vansteenkiste (eds.), *Artificial intelligence in numerical and symbolic simulation*, ALEAS, Lyon (FR), pp25-49 (1991).

[2] Laurent Buisson, "Reasoning on space with knowledge object centered representation", *Lecture notes on computer science*, 409:325-344 (1990).

[3] Laurent Buisson and Jérôme Euzenat, "A quantitative analysis of reasoning for RMSes", Proc. 6th internationnal symposium on methodologies for intelligent systems (poster session), Charlotte (NC US), pp9-20 (research report ORNL/TM-11938, Martin Marietta Oak Ridge National Laboratory, Oak Ridge (TN US)) (1991).

[4] Charles Forgy, "Rete: a fast algorithm for the many pattern/many objects pattern match problem", *Artificial intelligence*, 19:17-37 (1982).

[5] Hartmunt Freitag and Michael Reinfrank, "A non-monotonic deduction system based on (A)TMS", Proc. 8th ECAI, München (DE), pp601-606 (1988).

[6] Jérôme Euzenat, "Cache consistency in large object knowledge bases", Laboratoire ARTEMIS/Imag internal report (1990).

[7] Jérôme Euzenat and François Rechenmann, "Maintenance de la vérité dans les systèmes à base de connaissance centrée-objet", Proc. 6th AFCET-INRIA conference on "Reconnaissance des Formes et Intelligence Artificielle", Antibes (FR), pp1095-1109 (1990).

[8] Yoshihiko Otha and Katsumi Inoue, "A forward-chaining multiple-context reasoner and its application to logic design", Proc. conference on Tools for Artificial Intelligence, Herndon (US), pp386-392 (1990).

[9] François Rechenmann and Patrice Uvietta, "SHIRKA: an object-centered knowledge base management system", in: Alain Pavé, G. Vansteenkiste (eds.), *Artificial intelligence in numerical and symbolic simulation*, ALEAS, Lyon (FR), pp9-23 (1991).

[10] Murray Shanahan, "Incrementality and logic programming", in: Barbara Smith, Gerald Kelleher (eds.), *Reason maintenance systems and their applications*, Ellis-Horwood, Chichester (GB), pp21-34 (1988).