A CBR Integration From Inception to Productization

Lisa Purvis
Xerox Corporation
800 Phillips Road, 207-01Z
Webster, NY 14580
E-Mail: lpurvis@wrc.xerox.com

Abstract

Our case-based reasoning (CBR) integration with the constraint satisfaction problem (CSP) formalism has undergone several transformations on its journey from initial research idea to product-intent design. Both unexpected research results as well as interesting insights into the real-world applicability of the integrated methodology emerged as the integration was explored from alternative viewpoints. In this paper, the alternative viewpoints and the results that were enabled by these viewpoints are described.

Introduction

Although Case-Based Reasoning is a natural formulation for many problems, there are elements of the paradigm that prevent the methodology from being more widely applied. In particular, the problem of case adaptation is often considered to be one of the most difficult portions of a case based reasoner, because it encompasses many problematic characteristics: convergence upon a solution can be difficult to guarantee, widely varying case representations make it difficult to combine several cases, and domain-specific adaptation rules are often required, limiting the application of an adaptation methodology to one domain.

In order to address these challenges, a master-slave integration between CBR and CSP was developed, where a CSP algorithm was used to perform the CBR adaptation process.

In the process of implementing this integration and testing it on assembly sequence design problems, some reciprocating benefits for CSP were also discovered, that were enabled by the pairing of CSP with CBR. These results caused the integration to be reconsidered from the slave-master viewpoint, with CBR as the slave.

Re-examining the integration from the CBR-as-slave viewpoint highlighted a new opportunity for its application: using the methodology for dynamic constraint satisfaction problems (DCSP).

The DCSP approach has proven to be applicable in product software being developed to do a real-time planning task. The integrated approach for solving DCSP enables many of the desired product attributes.

This paper describes the lifecycle of the integration, as it was examined from the differing viewpoints, concluding with a description of the desirable product attributes that are enabled by the synergy of the two reasoning modes.

Related Work

Being one of the more difficult portions of case-based reasoning, adaptation has been recently receiving much attention from the CBR community. Recent systems that have addressed the multi-case adaptation issue are EADOCS (Netten & Vingerhoeds 1996), PRODIGY (Haigh & Veloso 1995), CAPlan (Munoz & Huellen 1995), and IDIOM (I. Smith & Faltings 1995). In EADOCS, each case addresses one feature of the new problem, and each case is used to adapt the corresponding solution feature. In PRODIGY, cases are replayed at specific choice points during the plan generation, and in CAPLan, the problem is analyzed into goals, the goals are used to retrieve cases, and each retrieved case replays its decisions. IDIOM uses a CSP during adaptation, with dimensionality reduction to eliminate constraint inconsistencies. In our work, the many matching cases are retrieved at one time from the case base during retrieval, and then these cases are all used simultaneously by the repair algorithm to find a solution to the new problem.

Case adaptability issues have been addressed in DEJA VU (Smyth & Keane 1995), where adaptability is assessed based on specially formulated adaptation knowledge, and is used to guide retrieval. Huang and Miles (Huang & Miles 1995) calculate the conflict propagation potential of a case to assess ease of adaptation during retrieval. Birnbaum et al. (L. Birnbaum 1989) and Leake & Fox (Leake & Fox 1992) index cases on the basis of their adaptability, avoiding cases with feature combinations that were difficult to adapt in previous problem-solving episodes. Portinale et al. (L. Portinale & Magro 1997) describe a technique called *Pivoting Based Retrieval*, in the context of case-based diagnosis, which is based on a tight integration between retrieval and adaptability estimation. In our work, where several cases are being simultaneously combined, assessing adaptability of each individual case during retrieval is difficult, as the global case interaction cannot be measured until all sub-cases participating in the global solution have been retrieved. Therefore, our contribution differs in that we assess adaptability *after* retrieval and matching.

Sqalli and Freuder (Sqalli & Freuder 1998) have done recent work on integrating case-based and constraint-based reasoning, where they formulate their problem (the interoperability of testing protocols in asynchronous transfer mode networks) as a CSP, and then use expert knowledge in the form of cases in order to deal with incompleteness and incorrectness. Weigel and Faltings (Weigel & Faltings 1998) use a CBR/CSP integration to achieve more flexible configuration design.

The Initial CBR/CSP Integration

The initial motivation for attempting to integrate CSP with CBR was to provide formalism to the CBR adaptation process, enabling it to apply across more than one problem domain.

To achieve this goal, the case-based adaptation process was formulated as a constraint satisfaction problem, where the cases were stored as CSPs (with constraints and solutions), and a CSP repair algorithm (the minimum conflicts repair algorithm (Minton, Johnston, & Laird 1992)) was used to perform the adaptation (Purvis 1995). This was a CBR-as-master approach, as CBR provided the overall problem-solving framework, with CSP helping with the adaptation process.

The integrated methodology was tested on both assembly sequence generation and configuration design problems, which showed that the integration was indeed domain independent, was guaranteed to converge on a solution if one exists, and provided a natural means by which to combine several cases (Purvis 1995).

Additional Results Obtained From the CBR-As-Master Methodology

Two research results emerged from the integration of CBR and CSP that were not initially targeted.

Reciprocating Benefit to CSP

The first additional result was that CSP received a reciprocating benefit from CBR: CSP solving efficiency was improved when starting with solutions from the case base. That is, providing the minimum conflicts repair algorithm with an initial solution that was composed of case solutions from the case base resulted in less backtracks during repair than did a simple greedy initialization. Details of the experimental results can be found in (Purvis 1995).

The second additional result was motivated by the observation that the improved performance for CSP was not universal in all case combinations. While solving efficiency overall was improved by starting with case solutions, there were some situations in which the case solutions did *not* provide better performance than did from-scratch problem solving.

These were situations where the retrieved solutions were difficult to combine, causing a repair of the entire problem, thereby disabling the retrieved case solutions from guiding the repair heuristic effectively (Purvis & Athalye 1997).

This discovery of the ineffectiveness of cases under some circumstances led to the realization of another benefit from the integrated approach: the ability to specify an adaptability criterion.

Adaptability Criterion

Adaptability of retrieved cases has been an elusive topic in CBR thus far, since it is difficult to tell before solving the problem whether it will adapt easily or not. At the same time, it is a very useful piece of information to have before expending adaptation effort only to later find that it is fruitless.

The experiments with assembly sequence generation showed that the minimum conflicts algorithm could not perform an efficient case combination when there were a large number of highly constrained, initially inconsistent edge variables (i.e. variables at the boundary between two cases) (Purvis 1995). The initial adaptability criterion based on these two factors is described in detail in (Purvis & Athalye 1997).

The adaptability criterion and the improved constraint solving efficiency were additional, originally unexpected results that were enabled by the combination of CBR and CSP.

The benefits of the integration for CSP solving led to further exploration of how the combined methodology might apply in a CBR-as-slave role.

How a CBR-as-Slave Viewpoint Provides Benefit to DCSP

From the CBR-as-slave viewpoint, rather than placing the CSP in a secondary role for adaptation purposes only, the CSP framework and the minimum conflicts algorithm are viewed as the guiding methodology, with CBR providing cases as input to the constraint solving.

With this change in viewpoint to CBR-as-slave, the applicability of the integrated approach for use with DCSP solving became apparent.

In a DCSP, not all constraints are known at the beginning of problem solving, but rather can be added or deleted as more about the problem becomes known. The DCSP is therefore a sequence of CSPs, where each one differs from the previous one by the addition or removal of some constraints (Dechter & Dechter 1988).

Such an addition or deletion of constraints can cause the current solution of the CSP to be invalidated, requiring re-solving the problem. The DCSP community has already recognized that some sort of reuse of the previous solution may be helpful during subsequent iterations of the DCSP in order to increase re-solving efficiency (Bellicha 1994; Verfaillie & Schiex 1994).

Examining the CBR/CSP integration from the CBR-as-slave viewpoint led to the possibility that as constraints change in a DCSP, cases may potentially help to increase re-solving efficiency.

Our current work revolves around the hypothesis that using an integrated CSP/CBR approach for DCSP enables more efficient and flexible dynamic constraint solving (Purvis 1997). Initial support for this hypothesis comes from examining how the difficulties in achieving efficient dynamic constraint solving can be alleviated by the abilities of CBR.

Combining Solutions Increases Re-Solving Efficiency

First, the case-based reasoning strategy of combining solutions is important for CSP, where time to solve can increase exponentially with the size of the problem. Using a CBR system that allows case combination enables the solutions of several smaller CSPs to be stored and re-used to efficiently solve larger problems. Thus, the experience of previous solutions can be reused even in the first iteration of a dynamic CSP. Those sub-cases that best match the larger problem can be retrieved and combined via the constraint-based adaptation described previously.

Increased Flexibility in Initial Solutions for DCSP

Second, a *similarity measure* can be defined that will search the case base full of previously solved problems for those that are 'most similar' to the new problem. By using a CBR framework for solving a DCSP, the CBR similarity measure can be used to free the DCSP from always having to begin with the solution found during the previous iteration of the DCSP. In some situations, the previous solution may not be the best starting point to enable efficient re-solving.

With a CBR component, the integrated methodology can use the CBR similarity measure to look for previous CSPs with more than just similar constraints. Other problem characteristics that are stored with the case can be taken into account, thus making it more likely that the retrieved solutions will be 'closer' to the final solution, a situation in which the minimum conflicts algorithm performs efficiently.

We are currently subjecting these initial intuitions to empirical testing on randomly generated CSPs.

Product Benefits from the Integration

The additional DCSP potential of the integration has highlighted the methodology's applicability in a real-world product. This product currently has a CSP software component that accomplishes a real-time planning task. The flexibility of the CBR/CSP integration has caused a closer look at how such an integration might enhance the product's planning abilities, and therefore favorably influence the product's marketable attributes.

Consistency of Operation

One immediately apparent benefit from the case-based DCSP approach is that solutions that have previously been found to be 'good' (and are therefore stored as cases in the case base), provide consistency to follow-on solutions that are made up of combinations of cases from the case base.

Consistency of operation is an important product attribute: customers do not expect erratic behavior from the product for no apparent reason. That is, when all externally visible characteristics and settings of a product are the same, the customer expects final output results that are consistent with those that were obtained the last time the product was used under these same circumstances.

The integrated approach for DCSP provides the opportunity to enable consistency of operation.

When starting with cases from the case base, the initial solution always consists of those cases that match the current problem situation. Thus, under the same problem circumstances, the same cases will be used to initialize the constraint solving algorithm.

When a greedy initialization is used, the current problem circumstances are not taken into account.

Thus, the same problem circumstances result in highly variable initial solutions on different runs.

The success of the minimum conflicts repair algorithm is attributed to its use of information from the initial solution to guide its search (Minton, Johnston, & Laird 1992). Providing the same initial solution guides the algorithm towards the same final solution.

Similarly, highly variable initial solutions will result in highly variable final solutions, as in the case of greedy initialization without the case base.

Scalability of Software

The high cost of software development is causing much focus on reuse in today's software organizations. When software is intended to be reused across different products, scalability of the software becomes important, so that differing product requirements can be met by the same software.

In terms of planning via DCSP, different product complexities influence the complexity of the DCSP problem solving. More complex products may cause the exponentiality of constraint solving to become prohibitive. Less complex products may not impose the same computational bounds, but rather a more stringent bound on memory usage and disk storage space.

Scalability that enables reuse of the integrated DCSP methodology is accomplished by allowing a flexible cooperation between CBR and CSP. On products whose memory and disk space requirements are too stringent to allow an extensive case base, CSP can operate alone, or with a limited CBR component with a small case base. For larger products, whose disk space and memory requirements are not stringent, but rather planning speed is more important, a more extensive CBR component can be used, that through its cases provides a means by which to control the computational complexity explosion as described earlier.

Such flexibility is only obtained due to the integration of CBR and CSP. Neither approach alone could satisfy all of these widely varying requirements.

Mass Customization

Todays products must compete in a highly competitive marketplace, with customers demanding and getting more and more features to fit their needs. A DCSP planning approach that contains a CBR component provides significant advantage over DCSP without CBR. The CBR component enables products to learn from their environment (by storing previous experiences as cases), and respond to each environment uniquely (by using the cases during planning).

The integrated methodology, therefore, is effectively enabling each customer to have a personalized product, without the product manufacturer spending extra time (and therefore also money) to hand-craft a customization.

Efficient Re-Planning in a Dynamic Environment

In a dynamic, real-time planning environment, the efficiency of replanning can become a significant product differentiator. The customer expects a fast, reliable, and quality response no matter what unexpected events might occur during product use. The customer will not be satisfied with a product that only works well under perfect conditions. Rather, the customer will prefer the product that can perform efficiently, consistently, and reliably no matter what dynamic events occur in the planning environment.

By providing the 'best matching' starting point for re-solving, and by enabling combination of existing solutions as described previously, fast re-solving is enabled by the integrated CBR/CSP approach for DCSP.

Future Work

There are many potential product benefits from applying the combined CSP/CBR methodology to do planning.

However, much work remains in order to provide empirical results supporting the benefits. The first planned empirical tests will be to run experiments to show that re-planning based on cases outperforms other DCSP approaches for re-planning from prior solutions.

Next, the existing data obtained from previous experiments will be synthesized to show that consistency of final solution is greater with the combined CSP/CBR approach than it is with the minimum conflicts algorithm alone. The previously done experiments with solving random binary CSPs with and without cases will provide input into this analysis.

Many of the desired product attributes described in the previous section compete with one another, or require a delicate balance in order to achieve their intended result. Additional future work will be to find the CBR and CSP characteristics that enable the correct balance of the two reasoning modes. Characteristics such as size of case base, content of cases, breakdown of cases, and overhead of matching will be studied, to see how each contributes to the desired product attributes.

Summary and Conclusion

We have described a CBR integration and the insights obtained as the integration was explored from alternative viewpoints. Our initial goal of defining a domain-independent adaptation methodology for CBR was achieved by combining CBR and CSP in a CBR-as-master approach. The resulting implementation caused a change in viewpoint to a CBR-as-slave approach, which in turn provided the ability to apply the integration for DCSP solving. The DCSP viewpoint was found to be applicable for a real-world planning task. The design of the integrated approach for planning has resulted in the realization that many of the desired real-world product attributes are in fact enabled by the integrated CBR/CSP methodology. This in turn provides evidence that work on CBR integrations has real value, particularly as these integrations begin to be applied to solve real-world problems.

References

Bellicha, A. 1994. Maintenance of solution in a dynamic constraint satisfaction problem. In *Artificial Intelligence in Engineering.*

Dechter, R., and Dechter, A. 1988. Belief maintenance in dynamic constraint networks. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, 37 - 42.

Goel, A. 1989. *Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving.* Ph.D. Dissertation, Ohio State University.

Haigh, K., and Veloso, M. 1995. Route planning by analogy. In *Proceedings of the First International Conference on Case Based Reasoning, Sesimbra, Portugal*, 169–180.

Hentenryck, P. V. 1990. Incremental constraint satisfaction in logic programming. In *Seventh International Conference on Logic Programming*, 189 - 202.

Huang, Y., and Miles, R. 1995. A case based method for solving relatively stable dynamic constraint satisfaction problems. In *Proceedings of the First International Conference on Case Based Reasoning, Sesimbra, Portugal*, 481–490.

I. Smith, C. L., and Faltings, B. 1995. Spatial composition using cases: Idiom. In *Proceedings of the First International Conference on Case Based Reasoning, Sesimbra, Portugal*, 88–97.

L. Birnbaum, G. Collins, M. B. e. a. 1989. A model-based approach to the construction of adaptive case-based planning systems. In *Proceedings of the Case Based Reasoning Workshop, Florida, USA.*

L. Portinale, P. T., and Magro, D. 1997. Selecting most adaptable diagnostic solutions through pivoting-based retrieval. In *Proceedings of the Second International Conference on Case Based Reasoning, Providence, RI, USA*, 393–402.

Leake, D., and Fox, S. 1992. Using introspective reasoning to guide index refinement. In *Proceedings of the Sixteenth International Conference of the Cognitive Science Society*, 313–318.

Minton, S.; Johnston, M.; and Laird, P. 1992. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence* 161–205.

Munoz, H., and Huellen, J. 1995. Retrieving cases in structured domains by using goal dependencies. In *Proceedings of the First International Conference on Case Based Reasoning, Sesimbra, Portugal*, 241–252.

Netten, B., and Vingerhoeds, R. 1996. Adaptation for conceptual design in eadocs. In *Proceedings of the ECAI'96 Workshop on Case Adaptation, Budapest, Hungary.*

Purvis, L., and Athalye, S. 1997. Towards improving case adaptability with a genetic algorithm. In *Proceedings of the Second International Conference on Case-Based Reasoning*, 403–412.

Purvis, L., and Pu, P. 1995. Adaptation using constraint satisfaction techniques. In *Proceedings of the First International Conference on Case-Based Reasoning, Sesimbra, Portugal*, 289–300.

Purvis, L. 1995. *Intelligent Design Problem Solving Using Case-Based and Constraint-Based Techniques.* Ph.D. Dissertation, The University of Connecticut.

Purvis, L. 1997. Dynamic constraint satisfaction using case-based reasoning techniques. In *Proceedings of the CP'97 Workshop on Dynamic Constraint Satisfaction.*

Smyth, B., and Keane, M. T. 1995. Experiments on adaptation-guided retrieval in case-based design. In *Proceedings of the First International Conference on Case Based Reasoning, Sesimbra, Portugal*, 313–324.

Sqalli, M., and Freuder, E. 1998. Integration of csp and cbr to compensate for incompleteness and incorrectness of models. In *Proceedings of the AAAI98 Spring Symposium on Multimodal Reasoning, Stanford, CA, TR SS-98-04.*

Verfaillie, G., and Schiex, T. 1994. Solution reuse in dynamic constraint satisfaction problems. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 307–312.

Weigel, R., and Faltings, B. 1998. Interchangeability for case adaptation in configuration problems. In *Proceedings of the AAAI98 Spring Symposium on Multimodal Reasoning, Stanford, CA, TR SS-98-04.*

Appendix

Integration Name/Category: COMPOSER, an integrated CBR/CSP system.

Performance Task: Assembly Sequence Generation, and more generally, Compositional Design tasks.

Integration Objective: To formalize the case-based reasoning process of adaptation using CSP techniques, to make adaptation more widely applicable.

Reasoning Components: Cases represented as CSPs, minimum conflicts algorithm to do the case combination.

Control Architecture: In COMPOSER, control is CBR-as-master. In follow-on work on dynamic CSP, control is CBR-as-slave.

CBR Cycle Steps Supported: Reuse and Revision. In the CBR-as-slave view, CBR supports the dynamic re-solving process by providing input cases from the case base.

Representations: Cases represented as CSPs.

Additional Components: Deductive retriever used for structural matching in COMPOSER.

Integration Status: COMPOSER is implemented and empirically evaluated. Follow-on work on Dynamic CSP is in proposed stage.

Priority Future Work: More in-depth theoretical analysis of the CBR-as-master approach, to determine whether there are any defining characteristics of the CSPs being combined that determine ease of combination. In the Dynamic CSP work, implementation and an empirical evaluation of the effect of seeding the re-solving with cases in comparison to existing DCSP solution reuse techniques.