Appropriateness of the Case-Based Approach in an application domain with multiple conflicts among Goals

Ricardo Luís de Freitas[1,2] , Gilberto Nakamiti[2], Carlos Miguel Tobar[1] , Dilermando Piva Jr[3]

Abstract. This work presents some of the important results we obtained on actions sequencing, conflicts avoiding, and past case information re-use, as well as to show the appropriateness of the proposed system, which was designed to a flexible manufacturing cell. Systems like this are known as case-based planning systems and constitutes a research topic within the Artificial Intelligence field which has received great attention from researchers world-wide, mainly for being potentially applicable in many areas.

1 MOTIVATION

One of the challenges for the complete automation of a flexible manufacturing cell is the implementation of computational systems, which have the ability to generate the correct actions sequence to be performed in the cell, given a part's specification. In this context, it would be desirable to explore Artificial Intelligence techniques to allow the complete transfer of this responsibility to the computational system. The first information is that a great difficulty to obtain the actions sequencing resides in determining the correct operations execution order, as well as in choosing the appropriate tool for each operation. The tool can produce undesirable damage in the part being produced. The numerous possible combinations of surfaces that can compose a part emphasize this difficulty. This way, the sequencing problem involves not only to define the best actions sequence, but also to obtain a satisfactory sequence in a reasonable time. Another characteristic of this application domain is its continuous need to be adjusted to new market requirements.

From these considerations, we tried to explore the adequacy and possible benefits of the use of a case-based planning system in the actions sequencing problem. We used Artificial Intelligence planning to provide the necessary flexibility to the system, and also a planning approach to explore the past cases re-use. To accomplish this, it was necessary the specification of an architecture that permitted the handling of the application domain information through a case-based planning system. Upon this architecture, we were able to implement a system prototype called PLANCEL-E [4], which was developed for a manufacturing cell composed by a CNC mill, a CNC lathe, and an industrial robot. PLANCEL-E is an evolution of PLANCEL system that, though had been used in manufacturing cells, were a general-purpose case-based planning system. Many case-based planning systems were studied, including: CHEF[5], GORDIUS [12], PLEXUS [1], SPA [7], CLAVIER [6], ARCHIE [9], CAPLAN [8], CASCADE [11], PRODIGY[13], ROBBIE [3], PARIS [2] and SINS [10].

[1] PUC-Campinas, Brazil;[2] Paulista University, Brazil,[3]Unicamp,Brazil
email: rfreitas@puc-campinas.edu.br; g_nakamiti@uol.com.br
tobar@puc-campinas.edu.br;piva@ieee.org

We applied the case-based approach with these systems to achieve ours. From these systems, we could achieve an initial model, further improved with new solutions to suit the FMC domain.

2 THE RETRIEVAL ALGORITHM

The retrieval process chooses among the existing cases, the one which is most suitable for the current situation. To this end, it searches the library case **B** for a case which contains the most promising plan of actions for the re-utilisation in the generation of a solution which allows the production of an object described by **O**. As input, the retrieval process requires the set of objectives ($\mathbf{O^n}$), which will specify the part to be produced.

This set of objectives is constituted by every surface of the specified part. After this, the user has to select one of the cases library superclass (**Spcl**). Then, all the classes related to the chosen superclass will be shown. A specific subclass can be chosen to conduct the retrieval process, or it will be performed with all the superclass. If a subclass is chosen (**Subcl** ≠ null), a past case ($\mathbf{b_i} \in \mathbf{B}$) containing a correct plan can be directly retrieved, through the use of the classes hierarchy as the retrieval key. When no subclass is chosen, the retrieval process has to check the existing restrictions and conflicts before continuing. So, the existing restrictions ($\mathbf{R^n}$) for the desired part are verified. To accomplish this, each objective ($\mathbf{S_i} \in \mathbf{O^n}$) is checked, verifying which tools ($\mathbf{F_i}$) are needed to attain it.

After analysing each surface ($\mathbf{S_i}$), the retrieval process defines the tools list ($\mathbf{F_i}$), which can be used on it. It looks for existing conflicts ($\mathbf{Cm^n}$) for each tool to produce the desired results. It uses the set of restrictions ($\mathbf{R^n}$) to verify if each objective ($\mathbf{S_i} \in \mathbf{O^n}$) has conflicts with neighbour objectives.

Then, the process looks for existing conflicts ($\mathbf{Cd^n}$) among different tools. It analyses if each tool actions to accomplish the desired objectives ($\mathbf{S_i} \in \mathbf{O^n}$) can other tools to accomplish different objectives ($\mathbf{S_j} \in \mathbf{O^n}$, $\mathbf{j \neq i}$). It uses the class hierarchy to reduce the search space. The retrieval process will define as candidate cases (**Cc**) the cases with similar superclass (and class). For each candidate case ($\mathbf{b_i} \in \mathbf{Cc}$), a degree of accomplishment of potential conflicts will be calculated (**Ei**) regarding to the part to be produced. The case with the greatest degree will be retrieved.

The output of the retrieval process will be the most appropriate case ($\mathbf{b_i}$) for the current situation. This case will be modified later by the adaptation process to completely satisfy the current demands. The exception to this procedure occurs when a specific subclass containing a perfect plan is chosen. In this case, no other process within the architecture will be needed because the plan stored in the retrieved case is already the final desired plan.

3 EMPIRICAL TESTS

In this work we intend to present some of the important results we obtained on actions sequencing, interaction goals avoiding, and past case information re-use, as well as to show the adequacy of the proposed architecture, which was designed to this application domain.

The existence of interactions goals in this application domain did not permit the use of existing similarity metrics based on goals, which are used on most current case-based systems. A new metric, to allow the reuse of good potential past cases, was needed. After a careful analysis, we decided to specify a new metric based on accomplished interaction goals, i.e., the case that solves more existing interaction goals of the current situation should be retrieved. Moreover, different levels of priority for the different kinds of conflicts were proposed. The different levels fit the varying difficulties of interaction goal accomplishment. To accomplish this, several situation tests were considered. The first test addressed the reuse of a past case that completely accomplished a new problem.

Obviously, the performance result was the same as the case retrieval time, since there was no need of adaptation. A second situation consisted in verifying the reduction in the execution time to produce a part with different dimensions of a previously produced part. The adaptation time of the actions plan, and the total plan generation time was measured.

The results obtained were also compared with the results of plan generation from different cases, chosen at random. The results obtained show a mean reduction rate of 82% in the total elaboration time from a previously generated plan with different dimensions.

Another test consisted in comparing the adaptation times when reusing cases with different levels of similarity related to the current one. Finally, the total times to achieve a plan reusing previous solutions and adapting them to the current situation were compared. The results show a performance improvement of more than 60% in the total times to achieve the new solution. In all the situations, we can see that the possibility of reusing more previous computational effort, present in the tests, leaded to significant reductions in the times needed to achieve new solutions. From these results, we could figure out that the performance improvement would be even larger when comparing to rule-based systems.

4 CONCLUDING REMARKS

The first contribution of this work is the definition of a format that permitted the representation of the application domain as a planning problem, and the appropriate handling of the difficulties, leading to the complete automation imposed by the domain. The format definition included the problem representation and handling as conflicts among goals. In addition, very interesting results were empirically obtained from PLANCEL-E. Through the results' analysis, it was also possible to get very important information about the planning system itself. Among the most important, we can cite:

- The adequacy of the similarity metric, which is based on problems to be, avoided (conflicts). This similarity metric can be used in application domains with multiple conflicts among goals, where each goal, if analyzed separately, does not help much to choose the potential cases;
- The reduction of context-dependent effects in the application domain through the generation and use of an appropriate representation format for the domain;
- The case-based approach appropriateness to allow the complete automation to generate the complete actions sequence to be executed in the manufacturing cell, generating correct plans in an acceptable time through the use of PLANCEL-E;
- The reduction of the solutions retrieval time through the use of an hierarchical organization of the stored cases;
- The elimination of solutions' redundancy through the re-use of the adaptation effort.

Finally, the development of this system leaded to a case-based planning framework, which can fit other application domains with similar characteristics.

5 BIBLIOGRAPHY

[1] Alterman,R. An Adaptative Planner. In Proceedings of the Fifth National Conference in Artificial Intelligence, Menlo Park, Calif.: AAAI, 65-69, 1986.

[2] Bergmann,R., Wilke,W. PARIS: Flexible Plan Adaptation by Abstractionand Refinement. ECAI 96 Workshop - Adaptation in CBR, Budapest, 1996.

[3] Fox, S., Leake,D. Modeling case-based planning for repairing reasoning failures. Proceedings of the 1995 AAAI Spring Symposium in Representing Mental States and Mechanisms, 31-38, 1995.

[4] Freitas,R.L., Nakamiti,G., Piva,D., Prado,J. SIMI: a Automated Environment for the Intelligent Planning of Actions in Flexible Manufacturing Cells. Proceedings of the IEEE SMC, Washington. 3470-3475, October 2003.

[5] Hammond,K.J. Case-Based Planning: Viewing Planning as a Memory Task. Academic Press, Cambridge, Mass., 1989.

[6] Hennessy,D.; Hinkle,D. Applying Case-Based Reasoning to Autoclave Loading. IEEE Expert, 14-20, October 1992.

[7] Kolodner, J.L. Improving human decision making througth case-based decision aiding. AI Magazine, v.12, n.2, p.52-68, 1991.

[8] Munhoz-Ávila,H.; Hendler ,J. Coonversational Case-Based Planning. New Review of Applied Expert Systems, 5, 1999.

[9] Pearce, M.; Goel, A.; Kolodner, J.L.; Zimring, C.; Sentosa, L.; Billington, R. Case-Based Design Support: A Case Study in Architectural Design. IEEE Expert, 14-20, October 1992.

[10] Ram,A.,Francis,A. Multi-Plan Retrieval and Adaptation in na Experience-based Agent. In Leake,D. Case-Based Reasoning: Experiences, Lessons and Future Directions, MIT Press, 167-183, 1996.

[11] Schmalhofer, F.; Globig, C.;Thoben, J. The reffitting of plans by a human expert. In: Joint Workshop Contemporary Knowledge Engineering And Cognition, 1o, Kaiserslautern, 1991, Proceedings. Berlin: Springer, 1992. p.115-24.

[12] Simmons,R. A Theory of Debugging Plans and Interpretations. In Proceedings of the Seventh National Conference on Artificial Intelligence, 94-99, 1988.

[13] Veloso,M., Carbonell,A., Perez,D., Borrajo,E., Fink,E. Blyte,J. Integrating Planning and Learning: The PRODIGY Architecture.Journal of Experimental Theoretical Artificial Intelligence, 7(1), 1995.