

COMBINING CASE-BASED REASONING AND PLANNING METHODS FOR INTELLIGENT SPARE PARTS PACKAGE PLANNING SUPPORT

Mehmet Göker Bettina Kühnapfel

Keywords: Package Planning, Case-Based Reasoning, Knowledge-Based Systems

1 Introduction

VAS 2000 is a project at debis SH for the Part Center of DaimlerChrysler. The Part Center currently stores and handles about 400.000 Mercedes-Benz spare parts. The majority of these parts has to be packed to be protected during storage and transport. The list of activities describing how and by which means the respective part has to be packed is called a packing instruction. Whenever a new car model is introduced or changes in marketing or ecological requirements occur, packing instructions for several thousand parts have to be generated in a short period of time. To define packing instructions manually is not necessarily very efficient and effective in such situations.

The goal of the VAS 2000 project is to create a software tool that assists in the generation and maintenance of packing instructions and ensures that the spare parts are packed according to the quality standards of DaimlerChrysler.

2 Solution Approach

A two phase approach was selected to generate packing instructions within the VAS 2000 system. In a first phase, a package planner who defines a similar package for similar parts is simulated by using Case-Based Reasoning [1]. If no adequate packing instruction can be found in the Case-Base, the system generates a packing instruction from scratch by using a planning algorithm.

2.1 Case-Based Package Instruction Generation

The package planners in the Part Center base most of their work on their experience and domain knowledge. When the description of a new part that has to be packed arrives, they search the database in which existing packing instructions are stored, retrieve a packing instruction that worked for a similar part, and adapt it to be usable for the new part. This

process is very similar to the Case-Based Reasoning (CBR) process model [1] and can be automated.

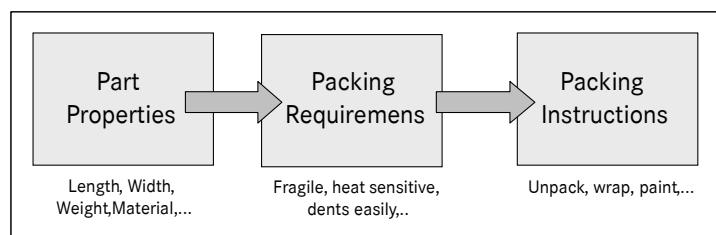


Figure 1 : Case Structure

The structure used to represent cases in the Case-Base is shown in Figure 1. The *part properties* contain data about physical, structural or organizational

specifications that are relevant for the package specification. They are used to compare parts and determine their similarity during the retrieval process. The *packing requirements* are properties that are derived from the part properties. They specify the requirements the package has to meet by stating the reaction of the part to environmental conditions. The packing requirements will be used in a later step to tightly couple the planning tool with the CBR-component of VAS 2000 and to enable the system to suggest adaptations. The *packing instruction* is a list of steps that have to be performed to pack the part. Each step does also list the material and devices that are needed to perform the respective packing step.

The CBR-component retrieves packing instructions for parts that are similar to the new part to be packed and presents these to the user. Since the cases in the Case-Base cover a broad range of parts, the Case-Based approach does provide good results for most queries. Due to the fact that the effort necessary to adapt an existing packing instruction of an old part to a new situation manually is not too high, we refrained from coupling the CBR-system with a planning system tightly. Even though approaches where cases are used as a starting point and are adapted to the new circumstances by planning systems exist [c.f. 2], we did not see an immediate need for this. However, both the case structure and the system architecture are designed in a way that permits a tighter coupling if desired. Autonomous adaptation is therefore not supported in the current version of VAS 2000.

2.2 Creating Packing Instructions from Scratch by Generative Planning

In the current implementation, the planning system generates a new plan from scratch if the solution of the CBR system was not acceptable. The prototype uses a STRIPS-Representation [3] to model operators and actions. The start state is the state in which the part is delivered, whereas the goal state is the way in which the part should be packed in the end. Searching the state-space, the program tries to find a sequence of operators (packing steps) that transform the start state to the final, desired state. Each packing step has a list of preconditions, a list of materials that are needed to perform this step, a list of properties to remove, and a list of properties to add to the state description. Each state description contains the name of the part to be packed and a list of properties that each operator can act on (Figure 2).

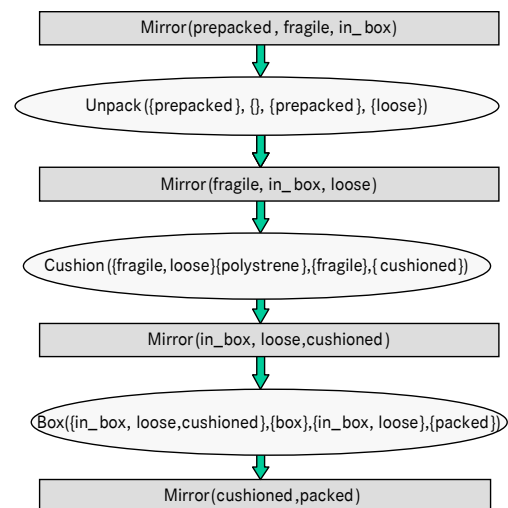


Figure 2: Creating Packing Instructions from Scratch

3 System Architecture

The architecture of the Integrated Package Instruction Generator component of VAS 2000 is shown in Figure 3. Currently only the Case-Based Reasoning component has been integrated into VAS 2000. The planning component does exist as a prototype.

The Integrated Package Instruction Generator consists of three basic modules.

- the Case-Based Reasoner,
- a Specification Transformation Component,
- and the Planning System

As described above, the CBR-component of the system retrieves packing instructions for similar parts based on the part specification. The Case-Base is directly linked to the database of VAS 2000. If the CBR-System is not able to find a similar part in the Case-Base, the part description is transferred to the "Specification Transformation Component". The purpose of

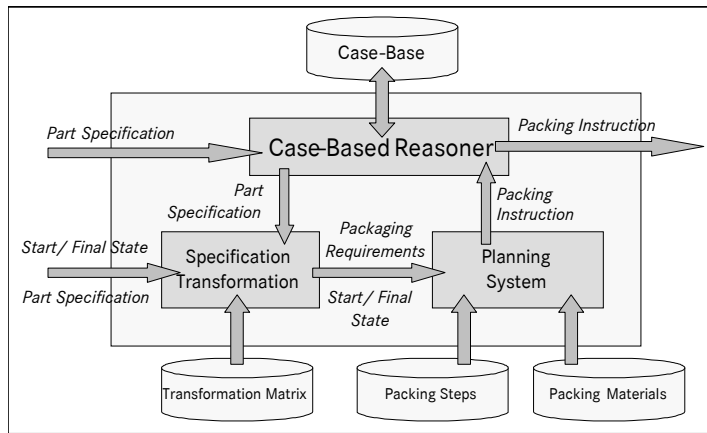


Figure 3: System Architecture of the Integrated Package

input data. The material that is available and the packing steps that can be used are kept in external databases and maintained by domain experts. Since the content of these databases varies between packing locations, keeping this information in external databases ensures that the system can be transferred between locations.

The new packing instruction is passed back to the CBR-component, presented to the user as a solution and, if applicable, learned.

4 Current Status and Further Developments

The Integrated Package Instruction Generator of VAS 2000 was implemented as a prototype to prove the concept first¹. This prototype showed, that the CBR component delivered an acceptable solution for the majority of parts and planning from scratch was only needed rarely. Based on the experience with this prototype, the CBR-component of the Integrated Package Instruction Generator of VAS 2000 has been implemented (Figure 4).

The system runs under Windows NT and is structured in three tiers: user interface, server and data. The user interface is implemented in JAVA. The server, which contains the CBR-component is in C++. The data is stored in an Oracle data base.

When a new part number is entered, the system retrieves the part specification from the parts database. Apart from physical information, this specification does also contain information regarding

Rel.	IblVpsText	Anzahl	AVO-ID	PkpNummer	IblVpsVstld	Vgh.
1	Auftrag Bearb.	1	10	0100	1	
2	Kartonage Ber.	1	20	0100	1	
3	Auftrag Zuteilen	1	60	0100	1	
4	Fremdvergabe S0	0	63	0100	1	
5	Lsk. Zuordnen	0	84	0100	1	
6	Schachteln mit	1	1.450	0100	3	
7	Teile in Schacht	1	1.500	0100	3	
8	Verschluß an Sc	2	4.700	0100	3	05
9	Etikettieren Sel	1	4.800	0100	3	02
10	Transporttaetig	1	6.200	0100	9	
11	Mehrzeit divers	0	6.300	0100	9	
12	Einlagerung na	1	6.400	0100	9	

Figure 4: CBR-Component of VAS2000

¹ The Prototype was implemented by Andreas Beil in the course of his Diploma Thesis at the University of Karlsruhe, Department of Computer Science in 1998.

lot sizes, recipient and if this is a 'repair kit'. Based on this information packing instructions of similar parts are retrieved and displayed.

Since the system operates in a client-server environment, and the part database resides on a central server, a method that puts a minimum amount of load both on the server and on the corporate network had to be implemented for similarity calculation and retrieval. In the current implementation the client requests similar parts from the server on an attribute to attribute basis. Retrieval is started with an attribute-value pair of the new part. If the number of cases that fall above a certain similarity threshold exceeds a fixed number, the value for an additional attribute is requested. After the number of similar cases falls below the set limit, these cases are transferred to the client and displayed to the user. The attributes that are going to be used in retrieval as well as their order can be determined by the user.

After a packing instruction has been selected, it can be edited and adapted to the new circumstances in the edit window shown in Figure 4. As a next step the implemented CBR-component will be integrated into the VAS 2000 system. Apart from generating packing instructions, VAS 2000 structures and controls the packaging process. As a further step, the planning component will be implemented, and coupled with the CBR-component.

5 Conclusion

To support the efficient generation, storage and re-use of packing instructions, a software tool (VAS 2000) has been developed. The approach is based on the assumption, that similar parts should be packed in a similar manner, i.e. by a similar packing instruction. CBR techniques have been applied to retrieve and re-use packing instructions that have been used for similar parts previously. If no adequate packing instruction can be found in the Case-Base, the system generates a packing instruction from scratch by using a planning algorithm.

Acknowledgements:

The authors would also like to thank Andreas Beil, Erwin Weiler and Annegret Zeller for their support during the project.

References:

- [1] A. Aamodt, E. Plaza., „Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches“, AICOM Vol.7 Nr.1, pp.39-59, March 1994
- [2] R. Bergmann, H.Munoz-Avila, M. Veloso, E. Melis, "CBR Applied to Planning", in M. Lenz, B.Bartsch-Spörl, H.D. Burkhardt, S. Wess (Eds.), "Case-Based Reasoning Technology, From Foundations ro Applications", Lecture Notes in Artificial Intelligence Vol. 1400, Springer-Verlag, Berlin, Heidelberg 1998
- [3] R.E. Fikes, N.J. Nilsson, "STRIPS: A new approach to the application of theorem proving to problem solving", Artificial Intelligence, Vol 2, pp189-208, 1971

Mehmet Göker
DaimlerChrysler Research and Technology
Machine Learning FT3/KL
P.O. Box 2360, D89013 Ulm
Germany
Phone: +49 731 5052856
Fax: +49 731 5054210
E-mail: mehmet.goeker@daimlerchrysler.com

Bettina Kühnapfel
debis SHE/KA
Stephanienstr. 102, D-76133 Karlsruhe,
Germany
Phone: +49 721 9104279
Fax: +49 721 9104298
E-mail: bkuehnap@debis.com