

VSKEP-EDITOR: A Visual Tool for Editing Reusable Skeletal Plans

Wu Zhaohui and Bernardi Ansgar *
German Research Center of Artificial Intelligence,
Postfach 2080, Kaiserslautern 6750,
Germany
email: {wu,bernardi}@dfki.uni-kl.de

Abstract

In order to reduce the acquisition task of expert knowledge about process planning in manufacturing, which are represented as skeletal plans in the PIM system, we propose to implement a visual editing tool called VSKEP-EDITOR. This tool not only can visualize the structure of skeletal plan and automatically check the syntax of skeletal plan when editing it, but also provides a taxonomic hierarchy of existing skeletal plans for reusing them to create new ones. These reusable skeletal plans can be adapted as the edited candidates to greatly reduce the task of knowledge acquisition.

1 Introduction

The PIM [1][2] system (Planning in Manufacturing) is a knowledge-based Computer Aided Process Planning (CAPP) prototype, in which a group of representation languages and formalisms are provided to fill the gap between

*The work is as part of the ARC-TEC project, supported by BMFT under grant ITW 8902 C4.

CAD and CAM. In these formalisms, SKEP_REP is a language for representing expert knowledge about process planning in manufacturing. Based on these expert knowledge in the form of SKEP_REP, the proposed PIM system can generate a production plan by performing a sequence of abstraction, selection and refinement. Therefore, Crafting these knowledge into PIM becomes more important.

After having analyzed the process planning in manufacturing and its presentation language SKEP_REP[3], we found the following features:

- There exists great redundancy among these skeletal plans and some new skeletal plans are acquired only through limit modifications on these existing skeletal plans.
- These skeletal plans formalized by the SKEP_REP language are closely related to their associated manufacturing feature structure[4], the visualization of these feature structure makes expert well understand about these knowledge.
- The SKEP_REP formalism is a language with a restrict terminology and a simple syntax template.

Therefore, in order to effectively reduce the acquisition task of expert knowledge in the form of SKEP_REP, we propose to implement the system called VSKEP-EDITOR (Visual SKEletal Plan EDITor), and to integrate the proposed tool VSKEP-EDITOR into the Common Lisp based PIM system, we adopt the following technical strategies or principles:

- to implement it by using the mature and powerful user interface programming facility CLIM. The CLIM (Common Lisp User Interface Management System) is a Advanced User Interface development system in Symbolics[5][6], which provides a group of facilities about program interface controlling, user input and formatting output to help users to approach the user interface of his/her application programs. These facilities will greatly reduce the developing task of VSKEP-EDITOR.
- to analyze the skeletal plan representation formalism SKEP_REP and to find out the template of representation and the visual structure of its feature structure. The simple version of SKEP_REP is described in the section 2.

- to investigate the organization structure of those existing skeletal plans for effectively reusing them to create a new skeletal plan. A taxonomic hierarchy of skeletal plans is shown in the section 3.

We are to introduce the architecture and corresponding mechanisms of this tool in section 4 and to illustrate its characteristics in section 5.

2 SKEP_REP: A Skeletal Plans Formalism

SKEP_REP a formalism for representing skeletal plan, which is used to describe expert knowledge about process planning on special part or feature of workpieces. A skeletal plan formalized in SKEP_REP consists of the following components:

- Op-Mode — to describe the operational mode of a skeletal plan.
- Special-Tree — to describe the feature structure of a skeletal plan.
- Constraints — to describe the local constraints of a skeletal plan.
- Actions — to describe the planning actions of a skeletal plan after having satisfied the feature structure and constraints. The actions part of the skeletal plan usually is a set of restrict lisp functions such as chucking, roughing, merging plans actions and so on.

3 A Taxonomic Hierarchy of Skeletal Plans

A taxonomic hierarchy of skeletal plans is built by effectively classifying. For one side, the classification of these skeletal plans can help expert to hierarchical browse all the skeletal plans which had been finished, and also assist him/her to fix the appropriate reused candidates of these skeletal plans in the hierarchy. On the other hand, we enable to add new skeletal plans into the hierarchy and to make these skeletal plans maintainable.

The taxonomic hierarchy of these skeletal plans are built from the partial order (\succeq) over these skeletal plans. This partial order is introduced through the partial relation (\succeq) over the feature structure contained in skeletal plan and the inclusive relation among the predicates of the local constraint in skeletal plan. We define the skeletal plan order as follows:

- Given two skeletal plan P-a and P-b, P-a contains of the feature structure P-a-FS and the constraint P-a-C, P-b contains of the feature structure P-b-FS and the local constraint P-b-C. the $P-a \succeq P-b$ iff $P-a-FS \succeq P-b-FS$ and $P-a-C \mapsto P-b-C$.

This partial ordering relation satisfy the following axioms:

- Reflexive: $U \preceq U$
- Transitive: If $U \preceq V$ and $V \preceq W$, then $U \preceq W$.
- Antisymmetric: If $U \preceq V$ and $V \preceq U$, then $V = U$.
- Top: $\top \succeq U$
- Bottom: $U \succeq \perp$

An illustrated taxonomic hierarchy of skeletal plans is shown in figure 2, where the "TOP-SKEP" is a generalization of the "SKEP-01".

4 Architecture

The architecture of the proposed tool VSKEP-EDITOR is illustrated in the figure 1. The basic components of this architecture are [7][8]:

- a controller – controls to read/interpret/execute the user input command.
- a command table – a extensive command table which provides the functions of VSKEP-EDITOR, and can be improved in the future version.
- three independent buffers for skeletal plans, the template buffer, the editing buffer and the reused buffer.
- a classifier — to classify these given skeletal plans into a hierarchy.
- a template-driven acceptor – to accept information from the template of skeletal plan and also to fill these items in the template.
- a lister – to input the user's command.

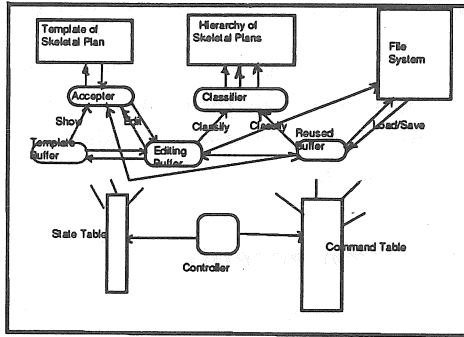


Figure 1: The Architecture of VSKEP-EDITOR

4.1 Independent Buffers for Skeletal Plans

As the proposed VSKEP-EDITOR supports various operations on skeletal plans stated in different types medium, these require us to implement various pools storing skeletal plans to promote the different requirements of command operations.

The separation of these individual buffers is to simply the complexity of implement and to enrich the functions of VSKEP-EDITOR. The follows listed are three kinds buffers for skeletal plan:

- **Editing Buffer** — to store these skeletal plans which have been edited. This buffer introduced is to support the edit of more than one skeletal plans and each of them can flexibly be added into the hierarchy of skeletal plans.
- **Template Buffer** — to store these skeletal plans which are used to be displayed or being edited. This buffer is used as a cache for the template of skeletal plan.
- **Reusing Buffer** — to store these skeletal plans which are loaded from the file system or classified from the other buffers, are illustrated in the hierarchy window and each of them can be selected as the reused candidate for editing a new skeletal plan.

4.2 A Classifier : Forms the Hierarchy of skeletal plan

The classifier illustrated in above architecture, is to classify skeletal plans in these buffers into the hierarchy. This means that it not only can classify the skeletal plans loaded from the file system, but also can classify these edited skeletal plans into the hierarchy.

4.3 Template-driven acceptor

In VSKEP-EDITor, we implement a template to describe the syntax structure of SKEP.REP. The template consists of three subtemplates, the feature structure template, the constraint template and the action template, which are extracted from the syntax of skeletal plan.

The acceptor can access the whole part of proposed template of skeletal plan and enables to display certain plan on this template and to show the text description of feature structure in the feature template in the visual way. The acceptor keeps checking on the syntax of input from three individual templates.

4.4 State-Variable Controller

The controller in the above mentioned architecture, is a simple command loop of read/interpret/execute, in which the controller is to maintain the user's command table and to control the execution of these commands in the command table.

4.5 User Interface

The user interface is a important component of VSKEP-EDITor. The configuration of user interface illustrated in figure 2, consists of the following five parts: (1) A Hierarchy Display Window for Skeletal Plan. (2) Two groups Command Menus. (3) Feature Structure Display Window. (4) Skeletal Plan Template Window, including the following three panes: Feature Structure Template, Constraint Template and Actions Template. (5) A Listener Window.

During using the VSKEP-EDITor tool, we can combine the command menu selection and character accelerator to input a user's command appear-

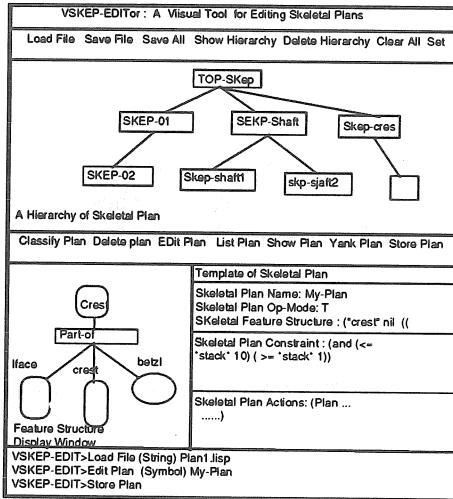


Figure 2: User Interface of VSKEP-EDITOR

ing in the listener window; and type-in or choose the output of representation by mouse to fill the arguments of the command such as the command " Edit Plan 'My-Plan ". The skeletal plan is generated by editing or modifying them within the template window.

5 Conclusion

VSKEP-EDITOR had been implemented in Commonlisp (totally 60 KB code) and as a tool in the PIM system and has the following characteristics:

- Hierarchical Skeletal Plans Reusable Edit – VSKEP-EDITOR can copy and yank the whole or part of these existing skeletal plans and form new one through limit modifications.
- Feature Structure Visualizable Edit – VSKEP-EDITOR is to visualize the feature structure of a skeletal plan automatically when editing it.

- Template-Driven and Syntax-Sensitive Edit — VSKEP-EDITOR provides a skeletal plan template driver to guide the operations of edit and a syntax automatic checker to cover the user's input.
- Friendliness of User Interface — VSKEP-EDITOR is to implement a command menu, command accelerator, button list, multiple windows, graphic formatting and terminological thesaurus to help expert interactively, friendly and conveniently edit these skeletal plans.

References

- [1] Bernardi A, etc:ARC-TEC:Acquisition,Representation and Compilation of Technical Knowledge, In Expert Systems and Their Applications:Tools, techniques and Methods,1991,pp133-145.
- [2] Legleitner, R., Bernardi, A,etc:*PIM: Skeletal Plan based CAPP*, Proceeding of Int' Conference on Manufacturing Automation, 1992.
- [3] Andres Becker:Analyse der Planungsverfahren der KI in Hinblick auf ihre Eignung fur die Arbeitsplanung, D-91-17, DFKI GmbH, 1991.
- [4] Klauck C., Bernardi A,etc:FEAT-REP: Representing Features in CAD/CAM. In IV Int' Symposium on AI:Applications in Informatics,1991.
- [5] Symbolics:Programming the User Interface, Book 10, 1990.
- [6] Symbolics:User Interface Dictionary, Book 11, 1990.
- [7] Zhaohui Wu and Bernardi Ansgar:User Guide of VSKEP-EDITOR Tool, DFKI Document, (Forthcoming), 1992.
- [8] Zhaohui Wu and Bernardi Ansgar:VSKEP-EDITOR: A Visual Tool for Editing Skeletal Plans, DFKI Document, (Forthcoming), 1992.