Using Artificial Intelligence Planning Techniques to Automatically Reconfigure Software Modules

Steve Chien[1], Forest Fisher[1], Helen Mortensen[1] Edisanter Lo[2], Ronald Greeley[2],
Anita Govindjee[1], Tara Estlin[1], XueMei Wang[3]

[1] Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109-8099
[2] Department of Geology, Arizona State University
P.O. Box 871404, Tempe, AZ 85287-1404
[3] Rockwell Science Center
444 High St., Suite 400, Palo Alto, CA 94301

Abstract. One important approach to enhancing software re-use is through the creation of large-scale software libraries. By modularizing functionality, many complex specialized applications can be built up from smaller reusable general purpose libraries. Consequently, many large software libraries have been formed for applications such as image processing and data analysis. However, knowing the requirements and formats of each of these routines requires considerable expertise - thus limiting the usage of these libraries by novices.

This paper describes an approach to allowing novices to use complex software libraries. In this approach, the interactions between and requirements of the software modules are represented in a declarative language based on Artificial Intelligence (AI) Planning techniques. The user is then able to specify their goals in terms of this language - designating what they want done, not how to do it. The AI planning system then uses this model of the available subroutines to compose a domain specific script to fulfill the user request. Specifically, we overview three such systems developed by the Artificial Inteligence Grou of the Jet Propulsion Laboratory.

The Multimission VICAR Planner (MVP) has been deployed for 2 years and supports image processing for science product generation for the Galileo mission. MVP has reduced time to fill certain classes of requests from 4 hours to 15 minutes. The Automated SAR Image Processing system (ASIP) which is currently in use by the Dept. of Geology at ASU supporting aeolian science analysis of synthetic aperture radar images. ASIP reduces the number of manual inputs in science product generation by 10-fold. Finally, the DPLAN system reconfigures software modules which control complex antenna hardware to configure antennas to support a wide range of tracks for NASA's Deep Space Network of communications and radio science antennas.

1 Introduction

The widespread use of software to automate a multitude of tasks has changed the way in which many tasks asre performed. One effect of this revolution has been an enormous increase in the complexity of the software that an average workr uses in an everyday fashion. Combined with this complexity is the enormous cost of producing needed software for a wide range of applications.

This considerable investment required to develop application specific software has led to the creation of large program libraries. These program libraries amortize the cost of software creation over a large and varied user base. By reducing the functionality of each individual piece into a modular chunk, reuse is encouraged and complex applciations can be built up from smaller building blocks; reducing the expense of software construction.

---

[3] This paper describes research conducted by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Other MVP team members include: Todd Turco, Christine Ying, Shouyi Hsiao, Darren Mutz, Alex Gray, Joe Nieten, and Jean Lorre. Other ASIP contributors include: Dan Blumberg(ASU), Anita Govindjee, John McHone(ASU), Keld Rasmussen(ASU), and Todd Turco. Other DPLAN contributors include Randall Hill Jr. and Teresa Yan.

However, these software libraries are not a panacea. Because of the complexity and diversity of the software libraries and their underlying execution environment, it requires substantial knowledge to know how to correctly use one of these software libraries.

For example, consider the use of image processing and data analysis liibraries by a scientist. In such an application, a scientist might want to use existing image processing and data analysis libraries to analyze newly available image data to discover patterns or to confirm scientific theories. Unfortunately, in order to perform this task, a complex set of operations is often required. First, before the data can be used it must often be reformatted, cleaned, and many correction steps must be applied. Then, in order to perform the actual data analysis, the user must manage all of the analysis software packages and their requirements on format, required information, etc.

Furthermore, this data analysis process is not a one-shot process. Typically a scientist will set up some sort of analysis, study the results, and then use the results of this analysis to modify the analysis to improve it. This analysis and refinement cycle may occur many times - thus any reduction in the scientist effort or cycle time can dramatically improve scientist productivity.

Unfortunately, this data preparation and analysis process is both knowledge and labor intensive. Consider the task of producing a mosaic of images of the moon from the Galileo mission (corrected for lighting, transmission errors, and camera distortions). Consider also that our end goal is to perform geological analyses - i.e., to study the composition of the surface materials on the moon. One technique used to do this is to construct a ratio image - an image whose values are the ratio of the intensity of the response at two different bandwidths (e.g., the ratio of infra-red response and visible green response). In order to correctly be able to produce this science product for analysis, requires knowledge of a wide range of sources including:

- the particular science discipline of interest (e.g., atmospheric science, planetary geology),
- image processing and the image processing libraries available,
- where and how the images and associated information are stored (e.g., calibration files), and
- the overall image processing environment to know how to link together libraries and pass information from one program to another.

It takes many years of training and experience to acquire the knowledge necessary to perform these analyses. Needless to say, these experts are in high demand. One factor which exacerbates this shortage of experts is the extreme breadth of knowledge required. Many users might be knowledgable in one or more of the above areas but not in all the areas. In addition, the status quo requires that users possess considerable knowledge about software infrastructure. Users must know how to specify input parameters (format, type, and options) for each software package that they are using and must often expend considerable effort in translating information from one package to another.

Using automated planning technology to represent and automate many of these data analysis functions enables novice users to utilize the software libraries to mine the data (p. 50 [Fayyad96]). It also allows users who may be expert in some areas but less knowledgable in other to use the software tools to mine the data.

However, our approach is not specific to science data analysis. The planning knowledge base is generically representing requirements of and interactions between software modules. Because there are many cases in which a subroutine library requires significant domain knowledge to operate - this approach has broad applicability. Indeed later in this paper we describe the application of these techniques to reconfiguration of antenna control software and elsewhere we describe application of thest techniques to assist in usage of complex semiconductor simulation and analysis software as part of a semiconductor design workbench [Brodley97].

The remainder of this article is organized as follows. First, we provide a brief overview of the key elements of AI planning. We then describe three planning systems which perform automated reconfiguration of software modules. We describe the MVP system - which automates elements of image processing for science data analysis for data from the Galileo mission. We then describe the ASIP system - which automates elements of image processing for science data analysis of synthetic aperture radar (SAR) images. Finally we describe the DPLAN system which reconfigures softtware modules to control complex antenna hardware to perform communications and radio science tracks.

The principle contributions of this article are twofold. First, we identify software tool reconfiguration as an area where AI planning technology can significantly facilitate program reuse. Second, we describe three systems demonstrating the viability and impact of AI planning on software reconfiguration process.

2 Artificial Intelligence Planning Techniques

We have applied and extended techniques from Artificial Intelligence Planning to address the knowledge-based software reconfiguration problem in general, and two applications in science data analysis (e.g., data mining) in specific. In order to describe this work, we first provide a brief overview of the key concepts from planning technology [4].

Planning technology relies on an encoding of possible actions in the domain. In this encoding, one specifies for each action in the domain: *preconditions*, *postconditions*, and *subactivities*. Preconditions are requirements which must be met before the action can be taken. These may be pieces of information which are required to correctly apply a software package (such as the image format, availability of calibration data, etc. Postconditions are things that are made true by the execution of the actions, such as the fact that the data has been photometrically corrected (corrected for the relative location of the lighting source) or that 3-dimensional topography information has been extracted from an image. Substeps are lower level activities which comprise the higher level activity. Given this encoding of actions, a planner is able to solve individual problems, where each problem is a current state and a set of goals. The planner uses its action models to synthesize a plan (a set of actions) to achieve the goals from the current state.

Planning consists of three main mechanisms: *subgoaling*, *task decomposition*, and *conflict analysis*. In subgoaling, a planner ensures that all of the preconditions of actions in the plan are met. This can be done by ensuring that they are true in the initial state or by adding appropriate actions to the plan. In task decomposition, the planner ensures that all high level (abstract) activities are expanded so that the lower level (subactivities) are present in the plan. This ensures that the plan consists of executable activities. Conflict analysis ensures that different portions of the plan do not interfere with eachother.

3 The Multimission VICAR Planner (MVP)

MVP [Chien96] partially automates generation of image processing procedures from user requests and a knowledge-based model of VICAR image processing area using Artificial Intelligence (AI) automated planning techniques. In VICAR image processing, the actions are VICAR image processing programs, the current state is the current state of the image files of interest, and the specification of the desired state corresponds to the user image processing goals.

The VICAR environment (Video Image Communication and Retrieval [5]) [LaVoie89] supports image processing for: JPL flight projects including VOYAGER, MAGELLAN, and GALILEO, and CASSINI; other space imaging missions such as SIR-C and LANDSAT; and numerous other applications including astronomy, earth resources, land use, biomedicine, and forensics with a total of over 100 users. VICAR allows individual processing steps (programs) to be combined into more complex image processing scripts called procedure definition files (PDFs). The primary purpose of VICAR is for encoding PDFs for science analysis of image data from JPL missions.

3.1 An Example of MVP Usage

In order to illustrate how MVP assists in VICAR planetary image processing, we now provide a typical example of MVP usage to ground the problem and the inputs and outputs required by MVP. The three images, shown at the left of Figure 1 are of the planet Earth taken during the Galileo Earth 2 flyby in December 1992. However, many corrections and processing steps must be applied before the images can be used. First, errors in the compression and transmission of the data from the Galileo spacecraft to receivers on Earth has resulted in missing and noisy lines in the images. Line fillin and spike removals are therefore desirable. Second, the images should be map projected to correct for the spatial distortion that occurs when a spherical body is represented on a flat surface. Third, in order to combine the images, we need to compute common points between the images and overlay them appropriately. Fourth, because we are combining multiple images taken with different camera states, the images should be radiometrically corrected before combination.

---

[4] For Further details on planning the user is referred to [4, Erol94]

[5] This name is somewhat misleading as VICAR is used to process considerable non-video image data such as MAGELLAN synthetic aperture radar (SAR) data.

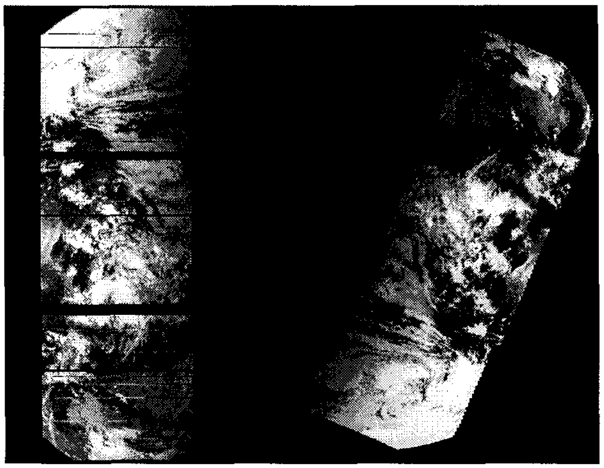


Fig. 1. Raw and Processed Image Files

MVP enables the user to input image processing goals through a graphical user interface with most goals as toggle buttons on the interface. A few options require entering some text, usually function parameters that will be included as literals in the appropriate place in the generated VICAR script. Figure 2 shows the processing goals input to MVP. Using the image processing goals and its model of image processing procedures, MVP constructs a plan of image processing steps to achieve the requested goal. Figure 3 shows the plan structure for a portion of the overall image processing plan.

radiometric correction
missing line fillin
no limbs present in images
display automatic nav residual error
display man nav residual error
mosaic images
pixel spike removal
uneven bit weight correction
perform automatic navigation
perform manual navigation
map project with parameters ...
smooth mosaic seams using DN

Fig. 2. Example Problem Goals

In this graph, nodes represent image processing actions (programs) and required image states to achieve the larger image processing goal. This plan is translated into a VICAR script which, when run, performs the desired image corrections and constructs a mosaicked image of the three input files. Figure 4 shows tha MVP-generated VICAR code corresponding to this subplan which performs image navigation [6] for a Galileo image. The finished result of the image processing task is shown at the right in Figure 1. The three original images now appear as a single mosaicked image, map projected with missing and corrupted lines filled in.

Thus MVP allows the user to go directly from high level image processing goals to an executable image processing program. By insulating the user from many of the details of image processing, productivity is enhanced. The user can consider more directly the processing goals relevant to the end science analysis of the image, rather than being bogged down in the details such as file format, normalizing images, etc.

MVP does not always fully automate this planetary imaging task. In typical usage, the analyst receives a request, determines which goals are required to fill the request, and runs MVP to generate a VICAR script. The analyst then runs this script and then visually inspects the produced image(s) to verify that the script has properly satisfied the request. In most cases, upon inspection, the analyst determines that some parameters need to be modified subjectively or goals reconsidered in context. This process typically continues several

---

[6] Image navigation is the process of determining the matrix transformation to map from the 2-dimensional (line, sample) coordinate space of an image to a 3-dimensional coordinate space using information on the relative position of the imaging device (spacecraft position) and a model of the target being imaged (e.g., the planetary body).

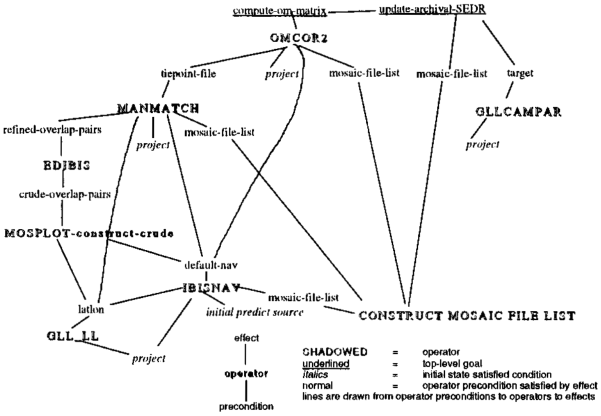
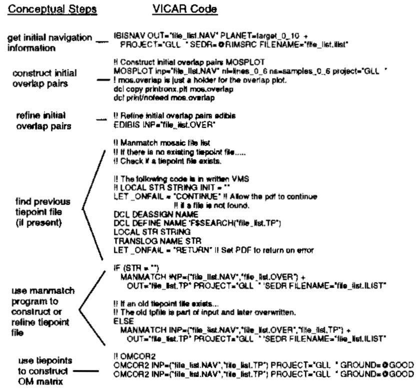


Fig. 3. Subgoal Graph for Manual Relative Navigation of Galileo Image Files



Conceptual Steps — VICAR Code

get initial navigation information

```
IBISNAV OUT="file_list.NAV" PLANET=target_0_10 +
  PROJECT="GLL  " SEDR=@RIMSRC FILENAME="file_list.ilist"
```

construct initial overlap pairs

```
!! Construct initial overlap pairs MOSPLOT
MOSPLOT inp="file_list.NAV" nl=lines_0_6 ns=samples_0_6 project="GLL  "
! mos.overlap is just a holder for the overlap plot.
dcl copy printronx.plt mos.overlap
dcl print/nofeed mos.overlap
```

refine initial overlap pairs

```
!! Refine initial overlap pairs edibis
EDIBIS INP="file_list.OVER"
```

find previous tiepoint file (if present)

```
!! Manmatch mosaic file list
!! If there is no existing tiepoint file.....
!! Check if a tiepoint file exists.

!! The following code is in written VMS
!! LOCAL STR STRING INIT = ""
LET _ONFAIL = "CONTINUE" !! Allow the pdf to continue
                         !! if a file is not found.
DCL DEASSIGN NAME
DCL DEFINE NAME 'F$SEARCH("file_list.TP")
LOCAL STR STRING
TRANSLOG NAME STR
LET _ONFAIL = "RETURN" !! Set PDF to return on error
```

use manmatch program to construct or refine tiepoint file

```
IF (STR = "")
  MANMATCH INP=("file_list.NAV","file_list.OVER") +
    OUT="file_list.TP" PROJECT="GLL  " 'SEDR FILENAME="file_list.ILIST"

!! If an old tiepoint file exists...
!! The old tpfile is part of input and later overwritten.
ELSE
  MANMATCH INP=("file_list.NAV","file_list.OVER","file_list.TP") +
    OUT="file_list.TP" PROJECT="GLL  " 'SEDR FILENAME="file_list.ILIST"
```

use tiepoints to construct OM matrix

```
!! OMCOR2
OMCOR2 INP=("file_list.NAV","file_list.TP") PROJECT="GLL  " GROUND=@GOOD
OMCOR2 INP=("file_list.NAV","file_list.TP") PROJECT="GLL  " GROUND=@GOOD
```

Fig. 4. Sample VICAR Code Fragment

iterations until the analyst is satisfied with the image product.

Analysts estimate that MVP reduces effort to generate an initial PDF for an expert analyst from 1/2 a day to 15 minutes and reduces the effort for a novice analyst from several days to 1 hour- representing over an order of magnitude in speedup. The analysts also judged that the quality of the PDFs produced using MVP are comparable to the quality of completely manually derived PDFs.

4 Automating SAR Processing

ASIP automates synthetic aperture radar (SAR) image processing based on user request and a knowledge-base model of SAR image processing using AI automated planning techniques. ASIP enables construction of an aerodynamic roughness image/map (z0 map) from raw SAR data - thus enabling studies of Aeolian processes.

The aerodynamic roughness length (z0) is the height above a surface at which a wind profile assumes zero velocity. z0 is an important parameter in studies of atmospheric circulation and aeolian sediment transport (in laymans terms: wind patterns, wind erosion patterns, and sand/soil drift caused by wind) [Greeley87, Greeley91]. Estimating z0 with radar is important because is enables large areas to be mapped quickly to study aeolian processes, as opposed to the slow painstaking process of manually taking field measurements[Blumberg95]. The final science product is a VICAR image called a z0 map that the scientists use to study the aeolian processes.

ASIP is an end-to-end image processing system automating data abstraction, decompression, and (radar) image processing sub-systems, and integrates a number of SAR and z0 image processing sub-systems. Using a knowledge base of SAR processing actions and a general-purpose planning engine, ASIP reasons about the parameter and sub-system constraints and requirements: extracting needed parameters from image format and header files as appropriate (freeing the user from these issues). These parameters, in conjunction with the knowledge-base of SAR processing steps, and a minimal set of required user inputs (entered through a graphical user interface (GUI)), are then used to determine the processing plan. ASIP represents a number of processing constraints (e.g., that only some subset of all possible combinations of polarizations are legal as dependent on the input data). ASIP also represents image processing knowledge about how to use polarization and frequency band information to compute parameters used for later processing of backscatter to aerodynamic roughness length conversion - thus freeing the user from having to understand these processes.

Figure 5 shows an aerodynamic roughness length map of a site near Death Valley, California generated using the ASIP system (the map uses the L band (24 cm) SAR with HV polarization). Each of the greyscale bands indicated signifies a different approximate aerodynamic roughness length. This map is then used to study aeolian processes at the Death Valley site.

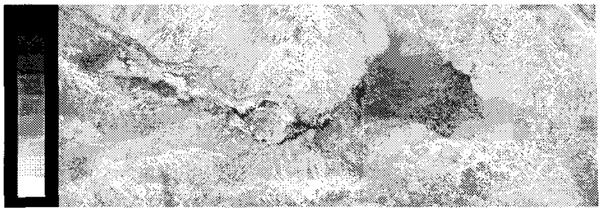


Fig. 5. Aerodynamic Roughness Length Map Produced Using ASIP

Since the ASIP system has been fielded, it has proven to be very useful in the use of generating aerodynamic roughness maps with three major benefits. First, ASIP has enabled a 10 fold reduction in the number of manual inputs required to produce an aerodynamic roughness map. Second, ASIP has enabled a 30% reduction in CPU processing time to produce such a map (by producing more efficient plans). Third, and most significantly ASIP has enabled scientists to process their own data (previously programming staff were required). By enabling scientists to directly manipulate that data and reducing processing overhead and turnaround, science is directly enhanced.

5 Antenna Control

The Deep Space Network Antenna Operations Planner (DPLAN) [Chien et al. 1996b, Chien et al. 1997a] is an automated planning system developed by the Artificial Intelligence Group to automatically generate antenna tracking plans to satisfy DSN service requests. In order to generate these antenna operations plans, DPLAN uses: the project generated service request (planning goals), the track equipment allocation (initial state), and an antenna operations knowledge base. DPLAN uses both hierarchical task network planning techniques and operator-based planning techniques to synthesize these operations plans. By allowing both operator-based and hierarchical task network representations, the antenna operations knowledge base allows a modular, declarative representation of antenna operations procedures. In contrast, consider the two non-AI alternatives proposed: operations script and an exhaustive library of plans. Neither operations scripts nor an exhaustive library of plans explicitly records the generality and context presumed by operations procedures. Planning representations explicit representation of such information should make them easier to maintain as DSN equipment and operations procedures evolve.

Through the use of AI planning techniques and the declarative representation used to model the operations of DSN antennas, DPLAN performs automated software module reconfiguration of control blocks used to build up control script. These control script are called Temporal Dependency Networks (TDNs). A sample TDN generated by DPLAN is shown in Figure 6. In order to correctly construct a TDN the planner must represent and reason about the requirements of individual soft modules and their interactions. The planner does this through the use of pre and post conditions for modules, and task decomposition rules as indicated earlier. Thus the planner can construct an appropriate TDN given the tracking goals, there by providing an interface for specifying 'what' task shall be performed vs. 'how' to accomplish that task.

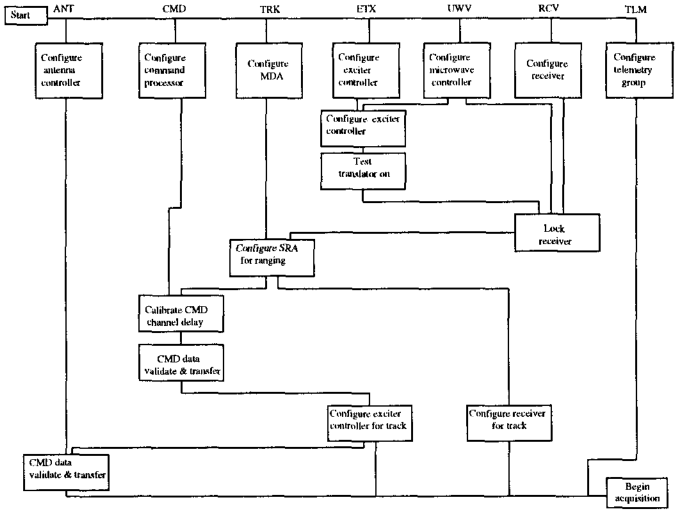


Fig. 6. Plan for precallibration for a Telemetry, Commanding and Ranging Track for a 34 meter Beam Waveguide Antenna

A similar approach is used on an other DSN project to automate antenna operations, the Deep Space Terminal (DS-T) [Fisher98]. DS-T also uses AI planning techniques to automnatically configure software modules. In the context of the DS-T the software modules are collections of sub-system directives which are then pieced together into larger sequences of commands to make up a track specific commanding script (similar to the VICAR scripts generated by MVP). These scripts are then executed to operate the DS-T antenna station for specific tracks.

6 Related Work

Related work can be broadly classified into: related image processing languages, related automated image processing work, and related AI planning work. In terms of related image processing languages, there are many commercial and academic image processing packages - such as IDL, Aoips,and Merlyn. Generally, these packages have only limited ability to automatically determine how to use different image processing programs or algorithms based on the problem context (e.g., other image processing goals and initial image state). These packages only support such context sensitivity for a few pre-anticipated cases.

Grimm and Bunke [1] developed an expert system to assist in image processing within the SPIDER library of image processing routines. This system uses many similar approaches in that: 1. it classifies problem types similar to the fashion in which MVP performs skeletal planning; and 2. it also decomposes larger problems into subproblems which MVP performs in decomposition planning. This system is implemented in a combination of an expert system shell called TWAICE (which includes both rules and frames) and Prolog. This very basic implementation language gives them considerable power and flexibility but means that their overall system uses a less declarative representation than our decomposition rules and operators which have a strict semantics [Erol94, 4]. on automating the use of the SPIDER library includes [3] which performs constraint checking and step ordering for a set of conceptual image processing steps and generation of executable code. This work differs from MVP in that: 1. they do not infer missing steps from step requirements; 2. they do not map from a single abstract step to a context-dependent sequence of image processing operations; and 3. they do not reason about negative interactions between subproblems. MVP has the capability to represent and reason about all 3 of these cases. Other work by Jiang and Bunke [2] involves generation of image processing procedures for robotics. This system performs subgoaling to construct image processing plans. However their algorithm does not appear to have a general way of representing and dealing with negative interactions between different subparts of the plans. In contrast, the general Artificial Intelligence Planning techniques used by MVP use conflict resolution methods to guarantee correct handling of subproblem interactions.

Perhaps the most similar planning and image processing system is COLLAGE [Lansky95] which uses AI planning techniques to integrate image processing in the Khoros environment. The COLLAGE planning differs from MVP, ASIP, and DPLAN in that COLLAGE uses solely the decomposition approach to planning while MVP, ASIP and DPLAN use decomposition based methods and operator-based methods. COLLAGE differs from MVP in the applications sense in that it focuses primarily on earth imaging applications in the Khoros environment, where MVP has focused on planetary applications in the VICAR environment.

7 Conclusions

This paper has described knowledge-based reconfiguration of data analysis and antenna control software using AI planning techniques. This represents an important area where AI planning can significantly enhance software usability processes. As evidence of this potential, we described three planning systems that perform automated software reconfiguration: the MVP system, which automates image processing to support Galileo image data science analysis; the ASIP system which automates production of aerodynamic roughness maps to support geological science analysis; and the DPLAN system which configures antenna software to perform communications and radio science tracks.

References

[Blumberg95] D. Blumberg and R. Greeley. "Field Studies of Aerodynamic Roughness Length." Jnl. Arid Environ. (1993) 25:39-48.

[Brodley97] C. Brodley, J. Fortes, M. Lundstron, S. Chien, S. Minton, C. Knoblock, "Distributed Infrastructure for Timely Proxy Collaboration and Metadata Integration," unpublished manuscript, 1997.

[Chien96] S. A. Chien and H. B. Mortensen, "Automating Image Processing for Scientific Data Analysis of a Large Image Database," IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8): pp. 854-859, August 1996.

[Erol94] K. Erol, J. Hendler, and D. Nau, "UMCP: A Sound and Complete Procedure for Hierarchical Task Network Planning," Proceedings of the 2nd International Conference on AI Planning Systems, Chicago, IL, June 1994, pp. 249-254.

[Fayyad96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, "From Data Mining to Knowledge Discovery in Databases," AI Magazine, Vol 17 No. 3, Fall 1996, pp. 37-54.

[Fisher98] F. Fisher, S. Chien, L. Paal, E. Law, N. Golshan, and M. Stockett "An Automated Deep Space Communication Station," Proceedings of the 1998 IEEE Aerospace Conference, Snowmass, CO, 1998.

[Greeley87] R. Greeley and J.D. Iversen, "Measurements of Wind Friction Speeds over Lava Surfaces and Assessment of Sediment Transport."G.R.L. 14 (1987):925-928.

[Greeley91] R. Greeley, L. Gaddis, A. Dobrovolskis, J. Iversen, K. Rasmussen, S. Saunders, J. vanZyl, S. Wall, H. Zebker, and B. White. "Assessment of Aerodynamic Roughness Via Airborne Radar Observations." 1991, Acta Mechanica Suppl.2, 77-88.

1. F. Grimm and H. Bunke, "An Expert System for the Selection and Application of Image Processing Subroutines, " Expert Systems, May 1993, Vol. 10, No. 2, pp. 61-74.
2. X. Jian and H. Bunke, "Vision Planner for an Intelligence Multisensory Vision System," Technical Report, University of Bern (extended version of a paper appearing in ICPR 1994).

[Lansky95] A. Lansky, M. Friedman, L. Getoor, S. Schmidler, and N. Short Jr., "The Collage/Khoros Link: Planning for Image Processing Tasks," Proc. 1995 AAAI Spring Symp. on Int. Planning Appl., pp. 67-76.

[LaVoie89] S. LaVoie, D. Alexander, C. Avis, H. Mortensen, C. Stanley, and L. Wainio, VICAR User's Guide, Version 2, JPL Internal Document D-4186, 1989.

3. K. Sakaue and H. Tamura, "Automatic Generation of Image Processing Programs by Knowledge-based Verification," IEEE Conference on Computer Vision and Pattern Recognition, pp. 189-192, 1985.
4. D. Weld, "An Introduction to least-commitment planning," AI Magazine 15 (19094) pp. 27-60.

This article was processed using the LaTeX macro package with LLNCS style