Web-Based Planning for Image Processing

Amy L. Lansky
Renaissance Research
1000 Fremont Ave. Suite Z, Los Altos, CA 94024
Contact: `lansky@renresearch.com` (415) 948-1983

January 23, 1996

1 Introduction

Over the past couple years, we have been applying a domain-independent planner, COLLAGE, towards automating Earth science image processing tasks. As reported in previous papers [5, 7], the planner has been successfully used to construct executable image processing data flows. Each component of a data flow corresponds to an image processing operation (e.g., application of a particular image filtering algorithm). Given a particular scientific goal (e.g. "determine vegetation content on the ground for each pixel in this image") and knowledge about the availability and nature of image processing algorithms and how they can be combined to achieve goals, the planner constructs a plan (data flow) which, after being executed, should yield the desired result. The planner can thus be viewed as a tool for automatic program synthesis, or alternatively, as a "component technology" - i.e. it constructs programs from given software components. Similar efforts include the Vicar [1] and Amphion [8] systems.

As also previously reported, we have found COLLAGE's unique *action-based planning* approach [3, 4, 5] to be well-suited to this and other realistic domains. Rather than trying to achieve state-based preconditions and goals, COLLAGE constructs plans by combining flexible task-decomposition techniques with various forms of constraint satisfaction. COLLAGE's constraint repertoire includes classic "CSP-style" constraints between plan variables as well as constraints on required temporal/causal relationships between actions.

Thus far, our efforts on the image processing domain have been focussed on enhancing COLLAGE's basic planning engine and developing modules for interfacing with image processing software packages. For example, we interfaced COLLAGE to the **Khoros** [9] image processing package so that COLLAGE plans can be translated into a form executable by **Khoros**. We also greatly enhanced COLLAGE's "CSP" facility, allowing for plan variables that are embedded within complex data structures (e.g., lists and records composed of subvariables). This was necessary for the image processing domain, where plan variables typically correspond to complex image-processing-algorithm parameters or lists of such parameters. The CSP facility can also now propagate numerical relationships between variables.

When we began developing the knowledge base for this domain, we also quickly recognized the need for software-engineering practices to organize and encapsulate domain knowledge so that it can be easily modified and reused. These needs and their importance to planning practitioners are discussed at length in [5].

As we now look forward to providing a tool that is truly effective for users, we are faced with issues that center more around *mode* of use (i.e. *how* the planner is used) rather than its innate planning capability. One of the primary goals of the larger data processing framework we are working within (Goddard's IIFS project [10]) is enabling access by a wide user base to image processing products. The average user may not have the sophistication of an experienced Earth scientist (and thus requires the automatic generation of image processing data flows). Of course, such a user may also not have access to COLLAGE, let alone sophisticated image processing toolboxes and the requisite hardware for executing image processing steps. Thus, in an ideal scenario, we envision image processing being conducted over the world-wide web. A remote user could make data product requests to a sophisticated image-processing web-site, which then automatically generates and executes image processing plans, yielding products that are sent back to the user.

Of course, such a framework bears similarity to other work on web-based access to data products – e.g. the work at ISI and Lockheed on planning for access to web-based (or otherwise distributed) data bases [2, 11]. What we are adding to these efforts is the idea that *the planning process itself can be an interactive process between a remote web-based user and planner.* The need for user/planner interaction over the web arises because of at least two factors:

- The knowledge base for this kind of planning will largely be developed by users rather than the planner-developers. For example, scientists will wish to augment the task-decomposition knowledge base with methods for generating their own particular kinds of image products, which they could then share with others.
- Scientific image processing is inherently iterative and interactive. Scientists usually need to repeatedly modify their image product requests based on results, and often wish to see intermediate results. This also brings up the possible need for dynamic, reactive planning capabilities.

2 Web-Based Planning

In this section we describe a high level architecture for a web-based planner. In the last section we discuss key issues relevant to this architecture.

Figure 1 depicts a possible architectural approach to a web-based planner for image processing. Along the perimeter are various modules with which users might directly interact. COLLAGE-specific modules along this perimeter include: an interactive user-manual/tutorial for training users who access the web site; a specification-building tool for adding domain knowledge or modifying domain knowledge within an existing corpus (domain knowledge would include task decomposition strategies, domain constraints, and fact-based knowledge about image processing – e.g., information about scientific quantities or about particular algorithms and packages); a tool for posting an image processing goal to COLLAGE, which would then trigger a specific plan-construction process; and a (possibly interactive) tool for display of plans as they are being constructed or executed. In addition to the COLLAGE-specific modules, the overall framework would include tools for direct access to image data bases and viewing of image products, and would possibly allow for direct interaction with image processing packages as well.

In a typical scenario, we envision that a naive user might post a simple goal, which would then trigger image-processing plan construction, execution of that plan, and return to the user of the resulting image product. Alternatively, a more sophisticated user might add new domain knowledge via the spec-builder, and, after posting a goal, modify the resulting plan during or after plan construction – or even during plan execution (such modifications might trigger more planning by the planning engine). A sophisticated user might also view intermediate results as the plan is being executed or choose to interact directly with an image processing package.

Of course, COLLAGE cannot support many of these capabilities right now. However, it can generate plans for simple goals and then translate them into data flows that are displayable and executable by the Khoros image processing system [5, 9]. Over the next year, we will begin development of the manual/tutorial module

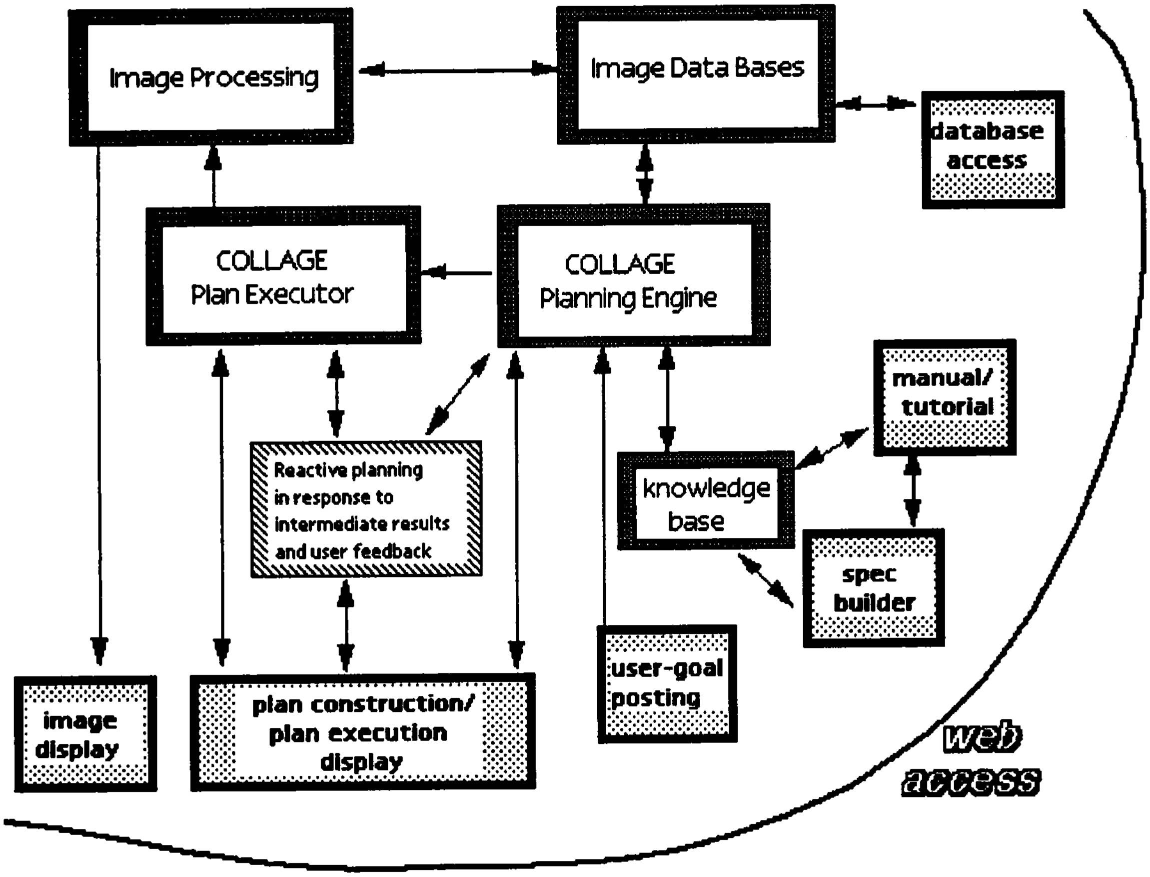

Figure 1: Web-Based Planning Architecture

and spec-builder module. We also hope to begin design work on extending COLLAGE to support simple reactive planning capabilities. In particular, we hope to support plan loops and conditionals via execution-time task-decomposition.

3 Issues

There are a myriad interesting questions and challenges posed by this architecture. Among them are the following:

- How interactive should the web-accessible modules be? For instance, the manual/tutorial and plan-display modules could be completely passive. But a more interactive approach, while significantly more difficult to implement, would enable a truly user-friendly and user-responsive framework.
- The spec-builder will probably be the most interesting module. There is room for work on "specification by example" (encouraging users to construct new domain knowledge based on existing specifications or templates), use of graphical tools to elicit specification structure and content, possibly some learning (e.g., for anticipating user requirements or desires), and correcting typical user mistakes. Many of these capabilities would be nontrivial to support over the web.
- Because of the inherently modular nature of the proposed architecture and its placement on the web, there is also natural potential for distributing its components. For example, the data bases and image processing hardware could be distributed. And because of COLLAGE's inherently localized (partitioned) planning representation and architecture [3, 6], the knowledge base and planning process could also be partitioned and distributed. Indeed, the COLLAGE planner could be viewed as a "distributed agent" generator, with the plan-executor as a kind of "agent-dispatch-monitoring-and-control" tool.
- If an image processing web-site is heavily used, how will it handle issues of scale? Distribution of workload again might be one answer.

4 Conclusions

Although our work on a "grand image processing architecture" is still preliminary, we have already demonstrated some useful capability. We have found it both fruitful and stimulating to guide our academic study of planning according to the needs of challenging domains and meeting user needs. This approach has certainly yielded a planning approach and technology ("action-based planning") that differs from the norm within the planning community. As planning applications mature and grow into frameworks with commercial potential, the nature of required planning technique as well as the natural relationship between planners and their users should be continually reexamined, expanded, and reworked. We believe this kind of dynamic provides the most exciting and useful context for work by planning researchers.

Acknowledgements

We acknowledge the significant contributions of present and past members of the COLLAGE project: Lise Getoor, Nick Short Jr., Mark Friedman, Scott Schmidler, and Andrew Philpot.

References

[1] Chien, S. "Using AI Planning Techniques to Automatically Generate Image Processing Procedures: A Preliminary Report," *Proceedings of the Second International Conference on AI Planning Systems*, Chicago, Illinois pp. 219-224 (1994).

[2] Knoblock, C. and Y. Arens, "Cooperating Agents for Information Retrieval," in *Pro-*

ceedings of the Second International Conference on Cooperative Information Systems, University of Toronto Press (1994).

[3] Lansky, A. "Localized Planning with Diverse Plan Construction Methods," NASA Ames Research Center, Artificial Intelligence Research Branch, Technical Report FIA-94-05 (1994). Under review for AIJ.

[4] Lansky, A. "Action-Based Planning" in *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems (AIPS-94)*, Chicago, Illinois, pp. 110 -115 (1994).

[5] Lansky, A., Friedman, M., Getoor, L., Schmidler, S., and N. Short Jr. "The COLLAGE/KHOROS Link: Planning for Image Processing Tasks," in *Proceedings of the 1995 AAAI Spring Symposium on Integrated Planning Applications*, Stanford University, Stanford, Californai (March 1995) and *Proceedings of the 1995 IJCAI Workshop on Artificial Intelligence and the Environment*, Montreal, Canada (August 1995).

[6] Lansky, A. and L. Getoor, "Scope and Abstraction: Two Criteria for Localized Planning," Proceedings of *IJCAI-95*, Montreal, Canada (August 1995).

[7] Lansky, A. and A. Philpot, "AI-Based Planning for Data Analysis Tasks," Proceedings of the Ninth Conference on Artificial Intelligence for Applications (CAIA-93), Orlando, Florida, pp. 390–398 (March 1993). Also appeared in *IEEE Expert Magazine*, Volume 9, Number 1, February 1994.

[8] Lowry, M., Philpot, A., Pressburger, T., and I. Underwood, "A Formal Approach to Domain-Oriented Software Design Environments," in *Proceedings of the Ninth Knowledge-Based Software Engineering Conference*, Monterey, California (September 1994).

[9] Rasure, J.R. and C.S. Williams, "An Integrated Data Flow Visual Language and Software Development Environment," *Journal of Visual Languages and Computing*, Volume 2, pp. 217–246 (1991).

[10] Short, N. et. al., "AI Challenges within NASA's Mission to Planet Earth," Workshop on AI Technologies for Environmental Applications, 1994 National Conference on Artificial Intelligence, July 31 - August 4, 1994, Seattle, WA, pges. 1 - 15.

[11] Toomey, C.N., Simoudis, E., Johnson, R.W., and W.S. Mark, "Software Agents for the Dissemination of Remote Terrestrial Sensing Data," in *Proceedings of the Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space (I-SAIRAS 94)*, Pasadena, California, pp. 19 - 22 (1994).