A Data Analysis Assistant

Amy L. Lansky

Recom/NASA Ames Research Center
Artificial Intelligence Branch
MS 269-2, Moffett Field, CA
94035-1000
(415) 604-4431
LANSKY@PTOLEMY.ARC.NASA.GOV

1 Introduction

This paper discusses the use of a domain-independent planner, COLLAGE, as a software assistant to Earth scientists working with remotely-sensed and ground-based data sets. The planner can be viewed as an *advisory agent* that helps a scientist to select appropriate data as well as to create a suitable plan for data-processing that meets stated scientific requirements [2].

Though we have worked on this domain for over a year, only recently have we come to view it as an instance of a much broader class of potential planning applications: helping humans to navigate through seas of software- and data-selection possibilities. In general, tasks that involve human interaction with visualization tools often manifest this particular kind of challenge. The human has very deep knowledge of their domain (e.g., the scientist knows about Earth science; the graphic artist knows what kind of image they are trying to produce). However, the tools available may be too vast or complex (e.g., there may be hundreds of possible data set options; there may be hundreds of data transform or image processing algorithms available). Thus, the human knows *what* they want to accomplish, but doesn't know *how* to use the software to accomplish it.

We believe there is great potential payoff in the development of software agents of this kind. Human experts desperately want the kind of help such agents could provide, and there is a high likelihood that such agents can be successfully implemented. This kind of domain has at least two other interesting characteristics:

- It would be almost impossible to imbue a software agent with enough deep knowledge about the domain to accomplish the desired task autonomously.
- It is feasible to imbue a software agent with the kind knowledge that a user *doesn't* have or doesn't want to be bothered with: what data and data manipulations algorithms are available; what functions these algorithms perform (at a high level of abstraction); and what usage constraints and requirements are attached to algorithms and data sets. For example, our Earth scientist experts currently make use of numerous data bases and at least two or three data analysis packages, each providing tens to hundreds of functions, with a variety of constraints on their use. The size and complexity of these data bases and packages, as well as their interactions, can make the data analysis task a logistical nightmare.

These two factors lead to a natural functional role for the kind of software agent we are developing. The planner will provide advice to the scientist about what data sets are available and what sequence of preprocessing algorithms may be appropriate for their task. However, it *does not* try to make data or algorithm choices that require deep scientific knowledge of the problem. Instead, the planner has a *dialogue* with the user, presenting useful information and plan options, interactively refining choices with the user, and performing constraint checking as appropriate, given its knowledge about domain requirements.

Thus, the role of our data analysis agent is to give the of level advice that the user wants and to stay well informed in order to provide that advice. Indeed, in this framework, good advice is more valuable to the scientist than autonomous "actions." Our software agent must "sense" or at least be aware of available data and algorithms, as well as feedback from the scientist. Our agent will "affect" its environment by providing advice to the scientist. Notice that this role is much deeper than that provided by a "smart interface." The kind of planning required is quite complex; scientists currently utilize human technicians to do much of what our software agent is being designed to provide.

The role of plan execution is also quite interesting in this domain. In some ways, "execution" may be viewed as the performance of actions that actually retrieve data and process it. Sometimes, the planner can execute these actions. In other cases, however, these actions must be performed by the scientist. This is because many image processing steps require human interaction – for instance, to select image points with the naked eye. In this latter case, the software planning agent merely *advises* the user on how to execute a step – what algorithm to use and possibly how to parameterize use of that algorithm.

There are also other forms of "execution" that could be considered in this domain. For example, consider the acts of making data set or algorithm *choices*. These choice-acts are made during the process of creating a data analysis plan rather than during actual data retrieval and processing. Often, partial execution (e.g., partial data set retrieval) must be performed in order to make further choices. As a consequence, the border between "planning" and "acting" in this domain quickly becomes blurred.

The rest of this short paper begins with a quick description of the data analysis task. Then, we provide a summary of current project status and recap two specific issues relevant to the symposium goals: planning vs. execution and agent survival.

2 The Data Analysis Task

The development and validation of Earth-system models and change-detection analyses require several data inputs that may be gleaned from a mixture of remotely-sensed images (taken by satellite instruments) and ground sources (e.g., meteorological readings, soil maps, and vegetation surveys). After data sets are retrieved and before they can be used, they must all be *registered* so that they lie within the same coordinate projection system and scale – i.e., all coordinate values must accurately correspond to one another. Unfortunately, the scientist's task of selecting suitable data and acceptably registering them is more difficult than it might seem. This process is often a burdensome and tedious portion of the scientific cycle that can consume over half of a scientist's time.

One reason is that required data is typically resident in disparate data bases and encoded in a wide variety of formats, densities, scales, and projections onto Earth's surface. Moreover, just as different kinds of information may be encoded in different forms, the *same* kind of information may exist in several different forms, may have been sampled in different ways, or may be derived through models. Thus, a scientist has many possible information sources to choose from, each associated with its own peculiarities and tradeoffs.

Once sources of information have been determined and data sets have been retrieved, scientists must register them. Unfortunately, heterogenous data types are often not directly comparable. For example, sparse temperature data collected by weather stations is usu-

ally not directly correlatable to satellite image data. Thus, a methodology is utilized that registers all data sets for a particular application to a common *base map*. Figure 1 depicts a high-level view of this process. First, a target coordinate system and scale is chosen. This target system is typically one that is similar to a majority of the data sets to be registered and that appropriately meets scientific and data-related constraints. Next, a base map of the study area is chosen that conforms to the target system. Then, all data sets are registered to this map. Depending upon the selected base map and the original form of a data set, required preprocessing steps may include geometric corrections, projection and scale transforms, radiometric corrections, atmospheric corrections, data restoration, interpolation, image enhancement, and ground control point selection (points that are used to achieve a correspondence between a data set and base map). Each of the steps depicted in Figure 1 would typically be composed of several substeps or processes. For each step, there are often a variety of possible algorithms, programs, and computational platforms. The choices made for each step must meet a variety of constraints that encode dependencies on and between registration steps. If poor choices are made, the registration process may introduce unacceptable distortions into the data. In some cases, registration may be impossible.

Consider the (simplified) registration plan depicted in Figure 2. Suppose that we have already selected and must now register two data sets – Thematic Mapper (Landsat) image data of Oregon and ground vegetation data for Oregon supplied by the US Forest Service in latitude/longitude coordinates. Our goal is to filter the image data through an equation that computes a vegetation index value for each image pixel and then plot these values against the ground-based vegetation values. First we select a target projection system of Universal Transverse Mercator (UTM) coordinates at a 30 meter scale and retrieve a suitable base map. Because latitude/longitude and UTM are both *universal* coordinate systems (there is a unique descriptor for each point on the surface of the Earth), the meteorological data is fairly readily registered using existing programs. Registering the Thematic Mapper data, however, requires the use of *ground control points*. Each ground control point is a physical location for which coordinate information is supplied from both the original data set and the base map. Using these coordinates, a transformation matrix can be computed that accurately translates all data set values into the target base map system. The challenge is finding adequate ground control point coordinates (both in number and accuracy) that are also uniformly distributed. If the points are skewed towards a certain portion of the study area, the transform matrix will yield unsuitably skewed results.

While there are some automated methods for finding ground control point coordinates, they are usually applicable only to limited data types and terrains. As a result, scientists generally select coordinates with the naked eye, utilizing recognizable features (e.g., cities and coastline features). However, if the original data set or base map does not contain enough discernable features, ground control point selection may be impossible and other options must be considered. For example, an alternate base map may exist for which adequate ground control points can be found. Or, a useable base map may exist in some *other* coordinate system that is then easily registered to the target system. One might also decide to choose an alternate target system or an alternate data source that has more identifiable features.

The data selection and registration process we have just described is full of compromises and tradeoffs, which also make it time-consuming and error-prone. There is intrinsic conditionality and interdependency between steps, often resulting in backtracking

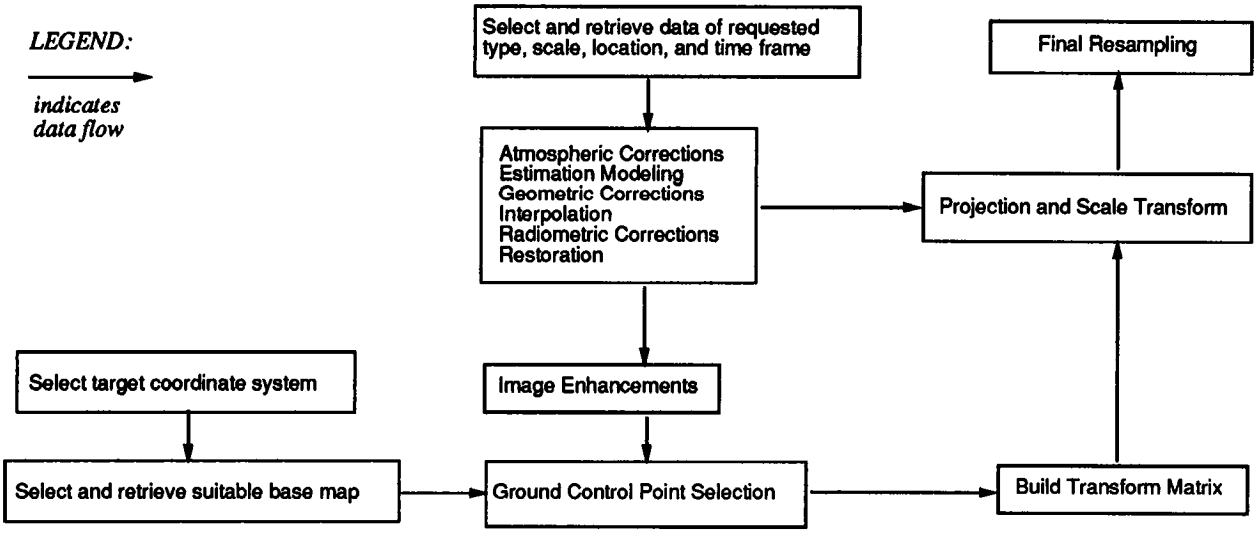

Figure 1: The Data Selection and Registration Process

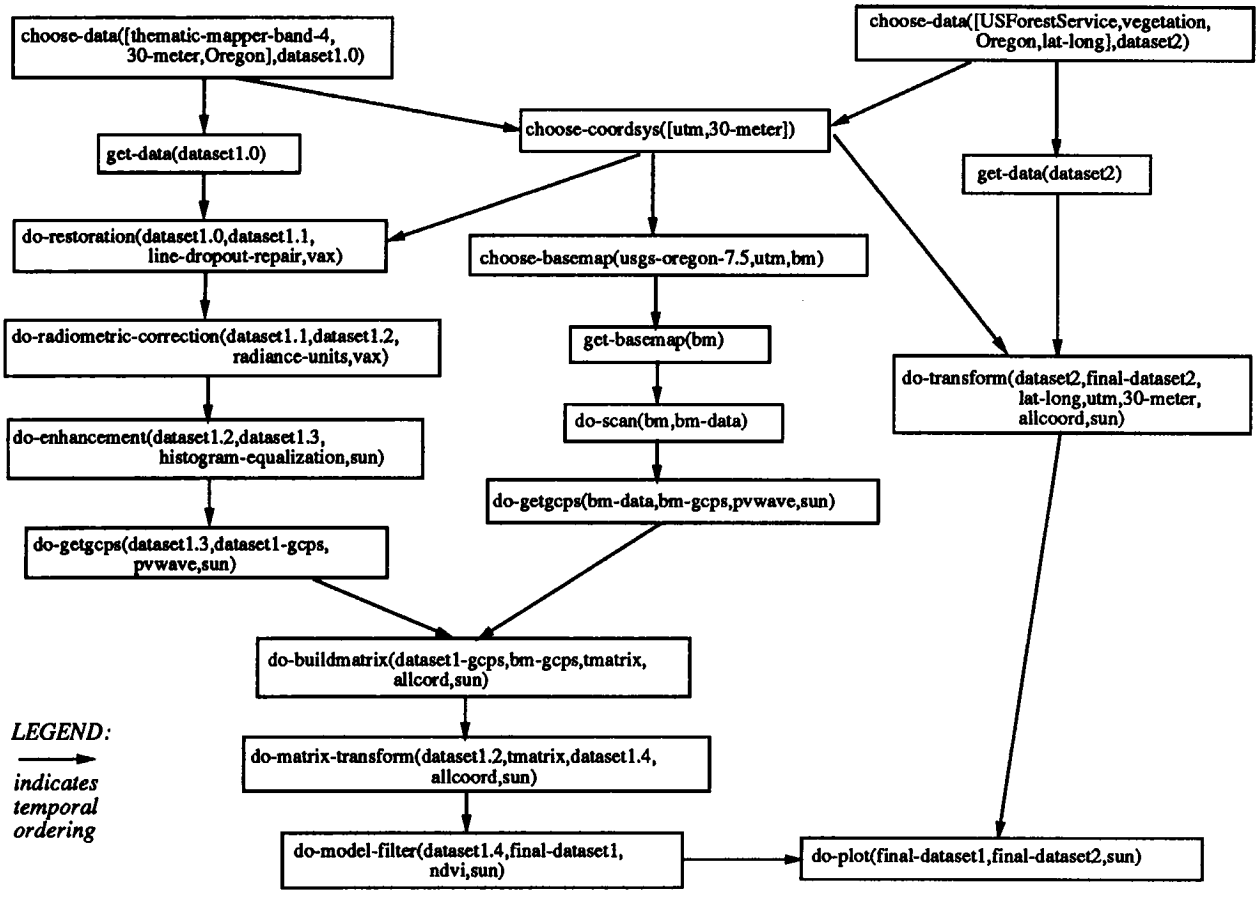

Figure 2: A Data Selection and Registration Plan

and replanning. In some cases, backtracking over critical decisions (e.g., changing the target coordinate system) may result in quite substantial changes to the overall plan. In other scenarios, failures or errors during execution may require portions of the plan to be modified "on the fly" (e.g., after data visualization, the scientist may realize that additional corrective transforms must be applied). Currently, scientists cope with the difficulties of this task by falling back on particular approaches they are familiar with, rather than those that are most suitable for a particular problem. As a result, they often end up using unsuitably flawed data sets. And because this process is rarely documented, it is quite difficult to diagnose the source of data distortions or to reuse previously successful plans.

All of these characteristics also make the data-selection and registration problem particularly amenable to automation. First, automation can help to speed up an otherwise tedious and time-consuming task. Automated reasoning also enables the exploration of a much more complete range of data selection and registration possibilities and much more thorough constraint and integrity testing. Finally, using an automated tool enables documentation and justification of data selection and registration choices, thereby allowing for the possibility of diagnosis and plan reuse.

3 Current Status

COLLAGE[1] is a non-traditional domain-independent planner that may be viewed as a general-purpose constraint-satisfaction engine [1]. In the COLLAGE framework, the term "constraint" is used very broadly — it is any type of requirement that the planner knows how to test and fulfill. Unlike the state-based encodings utilized by traditional planners, COLLAGE's action-based constraints describe domain requirements strictly in terms of desired action interrelationships action-parameter bindings requirements. The planner encompasses a wide variety of action-based constraint forms, each associated with constraint satisfaction algorithms that add new actions into a plan, decompose actions into subactions, impose ordering constraints, and constrain action-parameter bindings.

COLLAGE conducts its planning in a partitioned or *localized* fashion, searching a set of smaller (though possibly interacting) search spaces, each focusing on a subplan and a subset of the domain constraints. In the data analysis domain, these planning subproblems roughly correspond to the different data analysis subprocesses. In a sense, each of the local planning spaces may be viewed as an "agent," with the whole system being a community of coordinated planning agents.

Over the past year, we have encoded the the data analysis task in our planning constraint language and have extended the underlying COLLAGE planning framework and constraint library to meet the needs of this specification. We have also extended the system to include a static domain knowledge base that can drive and control aspects of the planning process. For this domain, the knowledge base includes facts about Earth's projection systems as well as information regarding algorithms that are available in geographic information system and image processing system packages commonly used by our scientist experts. We are currently working with these scientists to extend the domain knowledge base and create sufficient problem data to yield a set of planning problems for choosing and registering data for ecosystem models.

[1] **Coordinated Localized aLgorithms for Action Generation and Execution.**

4 Discussion

Planning vs. Execution

As we alluded in Section 1, this domain poses several questions about planning vs. execution for our software agent. As we began to write the domain constraints for this application and deepen our understanding of the role of our planner vis-a-vis the user, we began to see the line blurring between traditional notions of planning and execution. Much of the data analysis process must be planned in advance; for example, scientists would be loathe to order expensive data sets or waste their time performing tedious manually-intensive transforms unless they have created a data analysis plan that they are fairly sure will succeed. However, some forms of execution must take place *during* the planning process. For example, *choice actions* must be performed in order to determine a suitable target projection system. And some information about data sets must be retrieved *during planning* in order to enable reasoning about which algorithms are most appropriate to use.

Even allowing for some forms of planning-time "activity," some parts of the plan must be filled out or modified during actual data processing. For example, the ground-control-point selection process is often iterative – new points must sometimes be added, others deleted in order to yield the best registered image. These changes can't be determined until execution time, when an actual transform matrix is built and tried. Similarly, the most appropriate image enhancements for a data set often can't be fully determined until execution time, when the scientist can dynamically visualize those enhancements.

Unfortunately, this blur between planning and execution doesn't fit traditional notions of "reactive" planning either. This is because some pre-planning search-based reasoning *must* be done. Nor, obviously, does it fit more classical search-based preplanning. Instead, the desired planning behavior can be viewed as a *dialogue* between the planning agent and the user, who are involved in a *collaborative* effort. The software agent must be able to flow between classical deliberative reasoning, more dynamic forms of user-interaction and control over the planning process, and dynamic plan modification in response to the execution environment or user-directives.

For this reason, we have designed COLLAGE to enable a more fluid form of reasoning that we call "flexi-time" planning. The system already allows for some forms of actions (e.g., choices, data retrievals, interactions with the user) to be performed at "planning time." Soon, we hope to extend the primarily pre-planning framework of COLLAGE so that constraints can be triggered at any time relative to "execution." The COLLAGE constraint-triggering mechanism was intentionally designed to enable this kind of extension.

Agent Survival

Given the advisory role of our data analysis agent, the agent's survival is dependent on its level of utility to the user. Does it provide good, up-to-date advice? Is it easy to use? In the context of this domain, we are addressing these issues in at least two ways. First, we are placing all forms of domain knowledge that are relevant and understandable to the scientist in a domain knowledge base that is distinct from the planning engine and domain constraint specification. Unlike domain constraints, which drive the planning process, the knowledge base may be viewed as static domain- and/or problem-specific factual information. For this domain, the knowledge base consists of information about Earth projection systems, constraints on usage of specific data types, projections, and scales, information about available data transform algorithms, and problem-specific data selection

and registration goals. It also includes domain-specific functions such as those that determine the suitability of specific kinds of projections to certain areas of study. The planner uses the knowledge base by conditioning the constraint-satisfaction process on knowledge-base contents and by using the domain-specific facts and functions within it to impose constraints on plan variables.

Keeping the knowledge base distinct from the COLLAGE domain constraint specification and planning engine has several features that enhance utility:

- Planning functionality can be increased by extending the knowledge base rather than by extending the underlying domain constraint specification.
- The same domain specification can be used in numerous contexts with different knowledge bases.
- The knowledge base can be represented in a form amenable to viewing and extension *by the user.*

The last feature is critical since we, the developers of COLLAGE, cannot possibly gather all domain-relevant information for this application. New data bases and algorithms are always being developed within the scientific community. To be truly useful, the system must be extendable *by the user.* Thus, a critical aspect of the utility problem is domain knowledge capture, which we hope to facilitate by making incremental knowledge addition performable by the users themselves.

A second aspect of utility is *ease of use.* We hope to foster this through our development of COLLAGE's integrated user interface, COLLIE. The COLLIE user can visualize the growing plan, inspect properties of each action, relation, and binding, and understand the relationship between plan structure and domain constraints. Features are provided for viewing a graphical representation of the domain structure, visualizing the localized search process, and editing the domain specification and knowledge base. Eventually, we will extend COLLIE to include a better interface to the knowledge base and to allow users to modify the plan itself and to interact more directly with the constraint activation and search control mechanism.

Acknowledgements

We would like to thank our domain expert, Jennifer Dungan, for her time and patience. We would also like to acknowledge the contributions of past and present members of the COLLAGE project team: Lise Getoor, Andrew Philpot, Scott Schmidler, and Phil Chu.

References

[1] Lansky, A. "Localized Planning with Diversified Plan Construction Methods," NASA Ames Research Center, Artificial Intelligence Research Branch, Technical Report FIA-93-17 (June 1993). Submitted to AIJ, Special Issue on Planning.

[2] 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 to appear in a special issue of IEEE Expert Magazine, Winter 1994.