SEIDAM: A Flexible and Interoperable Metadata-Driven System for Intelligent Forest Monitoring

David G. Goodenough[1,2], Daniel Charlebois[3], A.S. (Pal) Bhogal[1], Andrew Dyk[1], Matthew Skala[1,2]

[1] Pacific Forestry Centre, Natural Resources Canada, Victoria, BC, Canada, V8Z 1M5
[2] Department of Computer Science, University of Victoria, Victoria, BC Canada, V8W 3P6
[3] Motorola, Chicago, USA
e-mail address: dgoodeno@pfc.forestry.ca ,fax: (250) 363-0775, telephone: (250) 363-0776

Abstract - The Advanced Forest Technologies Group at the Pacific Forestry Centre is continuing to develop a System of Experts for Intelligent Data Management (SEIDAM). SEIDAM manages large amounts of remotely sensed and GIS data and processes information for intelligent forest management and inventory updates. SEIDAM uses artificial intelligence (planning, case-based reasoning, software agents and machine learning) with previously captured domain expertise. SEIDAM uses a Prolog expert system shell called RESHELL. In order to manage and process natural resource information, SEIDAM relies on metadata that describes GIS data, field data and heterogeneous, multi-temporal remotely sensed imagery. In this paper, we discuss improvements to SEIDAM. The system is presently composed of a multitude of software agents that currently reside on a LAN. These agents are controlled by SEIDAM's main expert system and are synchronous in nature. By redesigning the interfaces between SEIDAM agents and the central system, SEIDAM will be able to operate in a distributed asynchronous manner across the Internet by taking advantage of new interchange protocols. For this initial implementation, we are concentrating on a suite of agents for automated analysis of AirSAR and AVIRIS data, beginning with the automated management of the hyperspectral and AirSAR meta data.

Introduction

SEIDAM (System of Experts for Intelligent DAta Management) integrates advanced information technologies. Its innovative approach to using AI (artificial intelligence - planning, case-based reasoning, machine learning), digital image analysis, geographical information systems and relational database systems has resulted in a system that:

- manages large amounts of heterogeneous data (optical and radar imagery, digital maps, relational database attributes),
- processes these data with minimal human interaction such that they can be presented to various users, typically forest managers, as products to support better decision making.

While the present approach has had many successes, there is still a need to further improve SEIDAM to achieve the vision of a distributed, intelligent information system which learns from experience and dynamically selects the best data sources to answer resource queries [1]. The first component of SEIDAM to be changed is the planner which creates the sequencing of agents to achieve a goal. The new PALERMO/TO reasoning system [2] does address some weaknesses of the first planner, PALERMO [3], but there is still a need for a more intelligent subgoal interleaving approach in order to produce better plans. For SEIDAM, better plans would result in a reduction in the amount of geospatial data processing that is carried out.

The second area where SEIDAM will be changed is its user interface. The original plan called for the use of TCL/TK as a scripting language to implement SEIDAM's GUI. The result, albeit appealing, has turned out to be a cumbersome component to implement and maintain particularly when completely new behavior must be introduced to the system. SEIDAM could be much improved if the interface were implemented in the same language as its reasoning systems (currently Quintus Prolog).

SEIDAM does currently use a primitive form of distributed processing. This must evolve now into a client/server architecture in order to allow it to take advantage of the wealth of data, knowledge and processing power distributed across the Internet. An architecture of this nature will allow the distribution of intelligent agents to client systems for knowledge acquisition and analysis.

Improving the Planner

The first planner, PALERMO, used goal-regression as its main search engine. Case-based reasoning was introduced as a means to allow the search engine to address problems that could not be solved by a linear search mechanism and to improve the performance of the problem solving. It succeeded on both counts. PALERMO solved problems in the one-way-rocket domain that could only be solved by a non-linear planner. It also was able to solve problems in the more complex logistics transportation domain and in the

0-7803-5207-6/99/$10.00 © 1999 IEEE.

SEIDAM domain. We also showed that PALERMO improved problem solving time by remembering each problem solved, and generalizing and re-using solutions.

PALERMO did have one significant problem: it stubbornly re-used nearly all the operators contained in the cases it retrieved when solving new problems. This behavior was mainly due to the fact that the type of analogy used in PALERMO was of a transformational nature rather than a derivational nature. In other words, PALERMO was designed to re-use the operators contained in a case rather than re-use the decisions it had made in selecting these operators. The result is that PALERMO can produce plans that contain redundant operators.

The experience gained in creating PALERMO showed that it was possible to improve the performance of a reasoning system while keeping the case-base small enough not to affect this improvement. PALERMO/TO was an attempt to draw on that experience. Two major shifts in design were implemented. First, the search engine for the planner was replaced by a much more powerful one. Second, the analogical paradigm that was applied to develop the case-based reasoning was derivational rather than transformational.

The new search engine is named TO (Total Order) and was developed by [4]. As the name of the planner implies, it creates plans that are totally ordered. However, it does so in a non-linear fashion. This simple change to the search mechanism used in PALERMO, insured that better plans were created by avoiding the introduction of redundant operators.

However, this type of search engine is far slower than a goal-regression engine. In fact in the logistics transportation domain as well as in the SEIDAM domain, complete test sets would take four to five hours to complete as opposed to the 30 to 45 minutes it would take for PALERMO to solve similar test sets. This was unacceptable. The first attempt at introducing analogy to address the performance problem was unsuccessful due to the use of transformational analogy. Because of the non-linear nature of the search engine, there was no simple way of introducing complete cases into the solution. Derivational analogy proved to be a better fit to this search engine. Each time the selection of an operator was required, a choice made during a previous problem solving exercise could be used for the new one.

The results were impressive. The new reasoning system performed roughly 30% better than the original system. It was able to solve some test sets in as little as 18 minutes and was always less than 30 minutes on a SUN Sparc 20. Moreover, because of how the search engine uses cases, redundancy is never an issue since only the decisions that are useful from the past case are used to solve the new problem.

There remains one problem. Because the planner is a total order planner, there is no subgoal interleaving. For example, if SEIDAM submits more than one goal to the system and the solving of one goal may provide part of the solution for another goal, the reasoning system can produce plans that contain unnecessary operators. For instance, there are several ways an image can be classified and the reasoning system might choose one method over another to solve a given goal. If for another goal during the same planning episode the system requires the same image to be classified via another approach, then the resulting plan will include both image classification processing threads although only one is necessary. To address this particular problem, the reasoning system must produce partial-order plans. Reasoning systems that produce this type of plan perform subgoal interleaving.

THE MAN-MACHINE INTERFACE

Great efforts were expended in developing a graphical user interface (GUI) for SEIDAM. During SEIDAM's gestation, it was decided that it was important to give the system a user friendly interface that was easy to use, portable to other platforms, easily maintainable and easy to grow. In choosing TCL/TK as a tool to develop the interface, it was believed that all of these goals would be achieved. Indeed, TCL/TK is easy to use and portable. However, the reality is that the latter two criteria have never really been met. Any time the interface's behavior must be adjusted to take into account a new parameter, an inordinate amount of time must be spent teaching the system about these changes. Also, the portability issue has become less and less of a consideration since future versions of SEIDAM will operate on either UNIX or NT.

The solution to the lengthy time required to implement window changes with TCL/TK is quite simple: implement the interface using the same programming language as SEIDAM itself. The problems for maintenance and for growing the user interface are mainly the result of having to create and maintain an interface between SEIDAM and TCL/TK. This interface is cumbersome, affects performance and is very difficult to maintain. Moreover, the windowing tools currently available within the Quintus Prolog environment are at least as easy to use as TCL/TK, allow the creation of every type of windowing object that SEIDAM needs and provide a significant performance improvement.

WEBIZING SEIDAM

The current implementation of SEIDAM can run applications on different computers thus allowing the system to take advantage of software packages available on a restricted number of platforms. The technique that has allowed SEIDAM to accomplish this is relatively simple: use software agents that reside in the same environment as

SEIDAM (i.e. Quintus Prolog) and have them logon to other computers as the same user currently running SEIDAM.

Each software agent is a RESHELL expert system [5] (the prolog/expert system environment for SEIDAM). When the software agents are activated, they spawn standard UNIX shells. From the shell's perspective, the software agents are the same as the users; i.e. the "users" type commands at the shell's prompt and the shell executes them. Thus, when the software agent executes the command "rlogin <host>" SEIDAM actually logs on to another computer and executes whatever software is required to complete the software agent's task.

This approach has proven very effective for the first SEIDAM release. However, it is somewhat restrictive. It only allows SEIDAM to perform processing on a LAN where each SEIDAM user has privileges across the network rather than across the Internet and does so in a strictly linear fashion. Given today's resources on the Internet, SEIDAM would become much more powerful if it were to take advantage of the services that can be found on other computers located in other networks of users.

This can be accomplished by re-designing SEIDAM's software agents to act as remotely or locally located servers to service a SEIDAM client. One could imagine an HTML browser submitting requests to servers able to perform the tasks currently carried out by the software agents. In fact, several computers could bid on the tasks SEIDAM would need to have carried out and the best bid would be awarded the contract.

To implement this type of architecture, it is necessary to design an interface between SEIDAM agents such that they do not have to be co-located. For maximum flexibility, the interface would have to comply with existing standards. Although TCP/IP or UDP/IP would be easy to implement, the current interface protocol that would ensure the greatest flexibility and the broadest accessibility would be an HTML interface. Scripts could easily be written and stored on the servers such that their activation would trigger SEIDAM agents. These agents would behave in essentially the same way they do today: the main difference is that they should be able to function in a non-interactive mode with the server.

META DATA

SEIDAM presently draws on two terabytes of remotely sensed data. SEIDAM will be used in a national system to monitor the forests of Canada [6]. Meta data in SEIDAM were originally held in approximately 20 relational tables in a relational data base (Ingres). Internet access to meta data led us to adopt the US FGDC standards for meta data. We added extensions to these standards to support an object-oriented meta data system. At the top level is the catalog meta data. The catalog level provides information about the spatial and temporal mission and general characteristics. The next level, the granule level, provides details of the image, sensor and platform [7]. The catalog level contains those parts of the metadata which can be used by the central directory level. For spatial data, the most important attributes for the catalog level are the geographical bounding coordinates appropriate to the site of interest, followed by the sensor name and the time frame of acquisition. Catalog-level metadata files each contain one or more Granule_Metadata_Online_Linkage fields, which contain local file addresses (URLs) pointing at directories of lower-level metadata files or individual lower-level metadata files. All catalog-level files will be stored in one central directory.

Each image file has low-level metadata associated with it, maintained in a separate metadata file. These granular metadata files are usually resident at the supplier's site. A "template" is maintained for each image data set, representing what granule metadata files look like for that set (e.g. TM, AVIRIS, AirSAR, etc.). These templates include the catalog-level metadata and any other fields that carry identical values in all the granules. They provide the redundant information specified by FGDC (since FGDC requires the duplication of catalog-level metadata into the granule metadata files.) The reason for creating these templates on a per-sensor basis is to simplify future changes. When the catalog-level metadata is updated, one can edit the template and automatically rebuild the granule metadata files, instead of editing every granule individually.

The granule-level metadata for fields that differ from granule to granule are stored in "merge files" suitable for use with Meta Merge. Meta Merge can combine the template files and merge files (both of which are abbreviated FGDC formal metadata text files) into FGDC-compliant metadata files suitable for use with Isite [8]. All meta data files must be verified by parsing with the FGDC "mp" tool for syntax checking as well as manually. The "mp" tool is configured for use with our extension syntax. We have implemented meta data files for Landsat Thematic Mapper, JPL AirSAR, and JPL AVIRIS data.

The meta data search procedure for this object-oriented structure is as follows. A) Search catalog-level files for a catalog record covering the desired area. Each record contains bounding coordinate information used for this level of the search. B) Link to the granule level. The catalog-level meta data file contains online linkage to the directory of granule-level files for its data set (for example, the "TM, 1993" set). C) Search the granule-level files for the specific granule record desired. D) Determine access to granule level. The granule metadata file contains a local file URL for automated access to the data; depending on data policy, external users may not be able to directly access the data. E) Create the FGDC-compliant meta data file. The granule-level meta data files are generated automatically from two kinds of input: templates (containing catalog meta data "inherited" from the catalog level, and granule-level metadata (that is the same for all the granules in the set)) and merge files which contain fields that differ from record to record. F) The granule-level metadata files contain redundant pointers to their corresponding catalog-level files (the converse of the links in "B").

Agents have been created which access the meta data automatically to select the best data to solve a particular problem. These agents operate over Canada's high speed information highway, Canarie. Intelligent access to meta data will be an essential component of our national system to monitor all the forests of Canada.

CONCLUSIONS

Design considerations for the evolution of an intelligent information system (SEIDAM) are presented. New directions include improving the planner to be capable of creating and using partial-order plans; replacing the man-machine interface; providing Internet access and distributed intelligent agents; and creating intelligent agents which access object-oriented meta data structures. The application focus of these improvements is to support the analysis of thousands of images of Canada's forests in order to provide accurate and timely reports of forest parameters, such as reforestation, afforestation, and deforestation.

REFERENCES

[1] D. G. Goodenough, D. Charlebois, A. S. Bhogal, and S. Matwin, "Automated Forest Inventory Update with SEIDAM," Proc. IGARSS'97, pp. 670-673, Singapore, 1997.

[2] D. G. Goodenough, D. Charlebois, A. S. Bhogal, and N. Daley, "An Improved Planner for Intelligent Monitoring of Sustainable Development of Forests," Proc. IGARSS'98, pp. 397-399, Seattle, Washington, USA, 1998.

[3] D. Charlebois, "A Planning System Based on Plan Re-Use and Its Application to Geographical Information Systems and Remote Sensing," in *Department of Computer Science*. Ottawa, Ontario, Canada: University of Ottawa, 1996.

[4] S. Minton, J. Bresina, and M. Drummond, "Total-Order and Partial-Order Planning: A Comparative Analysis," *Journal of Artificial Intelligence Research*, pp. 227-262, 1994.

[5] D. G. Goodenough, D. Charlebois, S. Matwin, and M. Robson, "Automating Reuse of Software for Expert System Analysis of Remote Sensing Data," *IEEE Trans. on Geos. and Rem. Sens.*, vol. 32, pp. 525-533, 1994.

[6] D. G. Goodenough, A. S. Bhogal, R. Fournier, R. J. Hall, J. Iisaka, D. Leckie, J. E. Luther, S. Magnussen, O. Niemann, and W. M. Strome, "Earth Observation for Sustainable Development of Forests (EOSD)," Proc. 20th Remote Sensing Symposium, pp. 57-60, Calgary, Alberta, 1998.

[7] A. S. P. Bhogal, D. G. Goodenough, D. Leckie, M. Skala, and S. Tinis, "SEIDAM Image Metadata Project - Extensions to FGDC Attributes," Pacific Forestry Centre, Natural Resources Canada, Victoria, BC, Canada, December 2, 1998.

[8] D. Bakewell, A. S. P. Bhogal, C. Burnett, D. G. Goodenough, D. Hill, D. Leckie, and S. Tinis, "An Infrastructure for Forestry Metadata Creation and Access," Proc. GIS'99, pp. 248-251, Vancouver, 1999.