



▶ Algorithm Optimally Orders Forward-Chaining Inference Rules

Requirements for exhaustive data-flow analysis are relaxed.

NASA's Jet Propulsion Laboratory, Pasadena, California

People typically develop knowledge bases in a somewhat *ad hoc* manner by incrementally adding rules with no specific organization. This often results in a very inefficient execution of those rules since they are so often order sensitive. This is relevant to tasks like Deep Space Network in that it allows the knowledge base to be incrementally developed and have it automatically ordered for efficiency.

Although data flow analysis was first developed for use in compilers for producing optimal code sequences, its usefulness is now recognized in many software systems including knowledge-based systems. However, this approach for ex-

haustively computing data-flow information cannot directly be applied to inference systems because of the ubiquitous execution of the rules. An algorithm is presented that efficiently performs a complete producer/consumer analysis for each antecedent and consequence clause in a knowledge base to optimally order the rules to minimize inference cycles.

An algorithm was developed that optimally orders a knowledge base composed of forwarding chaining inference rules such that independent inference cycle executions are minimized, thus, resulting in significantly faster execution. This algorithm was integrated into the

JPL tool Spacecraft Health Inference Engine (SHINE) for verification and it resulted in a significant reduction in inference cycles for what was previously considered an ordered knowledge base. For a knowledge base that is completely unordered, then the improvement is much greater.

This work was done by Mark James of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

The software used in this innovation is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-42003.

▶ Project Integration Architecture

All information of technological processes can be readily originated, manipulated, shared, propagated to other processes, and viewed by man or machine.

John H. Glenn Research Center, Cleveland, Ohio

The Project Integration Architecture (PIA) is a distributed, object-oriented, conceptual, software framework for the generation, organization, publication, integration, and consumption of all information involved in any complex technological process in a manner that is intelligible to both computers and humans. As used here, "all information" signifies, more specifically, all information that has been or could be coded in digital form. This includes not only experimental data, design data, results of simulations and analyses, organizational and financial data, and the like, but also sets of rules, computer programs, processes, and methods of solution.

In the development of PIA, it was recognized that in order to provide a single computational environment in which all information associated with any given complex technological process could be viewed, reviewed, manipulated, and shared, it is necessary to formulate all the elements of such a process on the most fundamental level. In this formulation, any such element is regarded as being composed of any or all of

three parts: input information, some transformation of that input information, and some useful output information.

Another fundamental principle of PIA is the assumption that no consumer of information, whether human or computer, can be assumed to have any useful foreknowledge of an element presented to it. Consequently, a PIA-compliant computing system is required to be ready to respond to any questions, posed by the consumer, concerning the nature of the proffered element. In colloquial terms, a PIA-compliant system must be prepared to provide all the information needed to place the element in context.

To satisfy this requirement, PIA extends the previously established object-oriented-programming concept of self-revelation and applies it on a grand scale. To enable pervasive use of self-revelation, PIA exploits another previously established object-oriented-programming concept — that of semantic infusion through class derivation. By means of self-revelation and semantic

infusion through class derivation, a consumer of information can inquire about the contents of all information entities (e.g., databases and software) and can interact appropriately with those entities.

Other key features of PIA include the following:

- Encapsulation of dimensionality and other semantically appropriate functionality;
- Enforcement of the dimensional nature of information (that is, something that is dimensional in nature cannot be accessed in a dimensionally-unaware manner);
- Exploitation of the object-identification facilities of the Common Object Request Broker Architecture (CORBA) to provide an object "address space" (defining the quantity of information that can be stored) that reaches to a practical infinity;
- Use of the object-etherealization and -incarnation facilities of the CORBA to make feasible the serving of a practically infinite number of objects;

- Use of file-system directory structures to make organization of, and access to, files reaching to a practical infinity feasible and efficient;
 - Provision of facilities to enable the storage of object state files in multiple locations, including locations accessible on the Internet through CORBA communication capabilities;
 - Provision of a distributed, lock-management system to efficiently provide for resolution of concurrent-access issues and, at the appropriate level, to apply per-user, per-object access controls;
 - Provision of a flexible event system so that attention or help can be automatically summoned when significant situations occur, significant information is found, and the like; and
 - Use of encryption to secure the transport of information when desired.
- The key benefits afforded by PIA include the following:
- Such software tools as graphical user interfaces, browsers, and search engines can be used to identify, view, manipulate, and share information from a practically unlimited number of resources;
 - Information can flow seamlessly and inerrantly between application pro-

grams and other information resources participating in the coordinated solution of a large problem;

- Because PIA is not discipline-specific, disparate technologies and their organizations can interoperate seamlessly when such needs arise;
- Flexibility in the integration of application programs is increased by enabling their authors to focus on the kinds of information needed rather than on the sources of the information;
- The identification of information propagation paths in integrated analyses is done by the machine, rather than by a person, thus reducing or eliminating the variability and occasional inaccuracy of human-directed efforts;
- Entire explorations of significant, planet-level technological problems can be captured and retained in an auditable, verifiable manner;
- The inherent parallelism of sibling activities is exploited to enable the concurrent investigation and analysis of planet-level technological problems by a practically infinite host of computers;
- PIA provides facilities by means of which it might become possible to automatically devise a method of solution of significant, planet-level technological problems and both the process of devis-

ing the method of solution and the method of solution itself could become a part of the body of available information;

- The potential ability to automatically devise methods of solution offers the possibility of developing adaptive, resilient systems that can “heal themselves” when individual elements fail or become unavailable;
- By providing such a mass of information in a single, cohesive environment, the automated search for recurrent patterns, themes, and the like is enabled; and
- PIA offers the potential for a new form of software product delivery in which the services of a software product are provided through a PIA-compliant server while the software, itself, remains closely held and managed by its owner.

This work was done by William Henry Jones of Glenn Research Center. Further information is contained in a TSP (see page 1).

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Innovative Partnerships Office, Attn: Steve Fedor, Mail Stop 4-8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-17961-1.