The G2 Development and Deployment Environment

Robert L. Moore

Gensym Corporation

Abstract

Gensym's G2 is an object-oriented development and deployment environment, combining rule-based and procedural reasoning, user interface graphics, database interface capabilities, dynamic simulation, and real-time execution into a single package. The applications of G2 typically have large knowledge bases, with tens of thousands of objects, and generally real-time execution is required. There are more than 2000 installations of Gensym's G2 in 28 countries. The on-line installations span the process industries, discrete manufacturing, aerospace, telecommunications, electric utilities and others. Primarily these installations have been implemented directly by plant engineers, who have defined the knowledge contained in the applications using G2's structured natural language and active-object graphics. This paper describes the design of G2, and provides examples of its use.

Knowledge representation

Objects

The knowledge representation in G2 is centered around objects, in a hierarchical structure with classes and inheritance. Objects in G2 can be permanent or transient, or they can be changed from one to the other. Objects can be created automatically by rules or procedures, as well as directly by the user using G2's structured natural language editor and active-object graphics. Objects may interact through connectivity, which can be done graphically by the user, or connections can be made or broken under rule or procedure control. Also object interactions may be defined by rules, procedures, relations, or analytic models. Typical G2 applications have from ten thousand to several hundred thousand objects.

The rules, procedures, models and other relationships in G2 are often expressed in generic form, across classes of objects. This allows a single statement of behavior to apply to many objects, and it is one of the most important sources of productivity. An example, expressed in G2's structured natural language might be:

"if the rate of change per minute of any temperature-sensor TS over the last 1 minute > 2 then invoke safety rules for each object connected to TS"

Behavior may be of various forms, such as rules or procedures about behavior over time, dynamic models, relationships, and others. Object behavior may change over time and according to conditions.

Connecting and cloning objects

G2 allows a schematic to represent objects and their interactions. The developer has the ability to define new objects, define attributes and graphically edit the schematic. For example, when a developer builds a new object, or calls for a new instance of an existing class, the new object can be graphically connected to other objects, and G2 understands this connection, and can reason and pass values over the connection. The developer can clone
an object, or a group of objects, and connect them. This extends the application, and it is a powerful way to create large applications quickly. This creating new objects and connecting them can be done by rules and procedures as well as by graphical operations.

Rules

G2 has several types of rules, which can be invoked in a variety of ways. G2 rules can be invoked whenever some event or change occurs, by forward or backward chaining, by timed execution, or by a variety of metaknowledge invocation techniques. For example, G2 can focus on a type of problem, and invoke the specific knowledge for diagnosing it. This ability to focus allows selective use of the appropriate knowledge in a large application.

The knowledge in G2 often deals with behavior over periods of time;

"if the rate of change of xx over the last 10 seconds..."

The behavior over time is often of greater interest than the current value. The knowledge itself also often has a time dependency, such recognizing a problem if nothing happens when something was supposed to happen, illustrating the need to have an understanding of time, and not just changes in data values. If nothing happens, it may be as serious as if some event happens.

Dynamic models

Dynamic models may represent knowledge about material flow, energy, chemical reactions, motion, or other response models. The state variable form of model permits non-linear time-varying representation.

\[ \frac{d}{dt} (\text{the concentration of oxygen in any fermentation reactor}) = \ldots \]

The successful plant applications are more commonly putting the engineer's knowledge on-line, as contrasted to the traditional expert system view of having knowledge engineers interviewing experts and packaging the operator's knowledge.

Procedures

There are advantages and disadvantages to purely inferential reasoning. One advantage is that knowledge can be added without regard to the order of the reasoning process. A disadvantage is that the order of reasoning may be important to control. Also procedural reasoning is generally much faster. So a full procedural reasoning, with object representation and other functionalities described above, is part of the G2 expert system. The procedural reasoning is commonly used for the "action" aspects of the system, such as batch control, generating intelligent displays, and such.

Real-time operation

Execution engine

The heart of G2's reasoning is a real-time engine for concurrently executing rules, procedures, models, and other items. The execution of a particular item can be driven by forward or backward chaining, a timed execution request, an external event or interrupt, or a focusing action. Everything has a priority, so critical items can take action first. Typically many lines of reasoning, models, and other activities are occurring concurrently. G2 has many features built-in to deal with real-world problems. For example, getting the best answer in a fixed time, reasoning when data is missing, time stamping data, and calculating the validity or probability of conclusions. Typical execution rates in G2 are thousands to tens-of-thousands of rules and procedures per second.

Data interfaces

The On-Line G2 licenses include a data server called GSI, which allows G2 to keep reasoning at full speed while data communications are underway. The many issues of high-performance, robust data communication are handled by GSI, such as buffering, hand-shaking, restore after break, automatic retransmit, and others. GSI can deal with timed requests for data in a look-ahead way, or with unsolicited data. GSI can handle ASCII strings, as an example for external message display. GSI has a "gateway" design, whereby any number of G2's can access the GSI extension, and a G2 can access any number of GSI extensions. This allows a G2 to access simultaneously data from databases, systems, or other sources, and networks of cooperating G2's can share such data. Typical data rates are hundreds to thousands of data values per second.

References

