Constraint Programming-based Configuration Cooperated with Rules-based Systems

03000500 アイログ(株) チェン ユウ CHENG Yu

1. Introduction

The configuration marketplace has changed significantly over the last year with the Internet and new demands for configuration technology for mass customization, specifically in the areas of personalization, eCommerce, eCRM, ePayment and eBilling. Web-based applications of configuration technology are becoming critical. At the same time, configuration is moving from a pure "push" technology where makers provide their customers with predefined products and services according to their profile, to a "pull" technology where customers can interact with the application and can tailor the product or service so it fits perfectly their needs. It is a basic requirement now that companies should be able to provide customized offerings to their customers, and thus configuration technology is becoming recognized as being key to providing this kind of functionality.

2 Configuration Problem and Solution Approaches

The task of configuration is to produce a set, or system, of interconnected components and choose a specific type for each component in the system. Component types are chosen from a predefined catalog. For example, a computer contains processors, and each processor must be implemented with a type of processor available in a computer parts catalog. The problem of choosing the component type can be treated as a classification problem, where the component instance must be positioned somewhere in the hierarchy of types. Besides, it should be possible to specify an optimization criterion such as the total price or the number of needed components. This requires the configuration engine to generate the best valid configuration for that particular criterion.

2.1 Various solution approaches

There basically exist three different paradigms for solving configuration problems.

- **Rule-based systems approach**

Traditionally, rule-based systems have been used to check feasible configurations. Rule-based systems were the first to tackle the configuration problem. They encompass both the structure of the product and the process for configuring it, mixing them in the same rules. This approach requires the rule database to grow with the parts catalog, and can lead to huge, barely manageable systems. As catalogs usually evolve quite rapidly, being updated weekly and even daily with some online services, maintaining such a database can quickly become an extremely difficult task. For this reason, rule-based configuration is currently losing ground.

- **Modeling-based approach**

In the model-based approach, the configuration problems are described as a specific mathematical model, or a simulation model, or a logical model in computer program. However, to solve the problem, the system must describe what to do and how to solve as procedural, like how to select or generate a particular part without violating constraints. It is usually a difficult task to make such kinds of solving procedures.

- **Constraint programming -based approach**

Configuration problems can be treated as a particular type of Constraint Satisfaction Problem (CSP). With Constraint Programming technology, it is essential possible and efficient to detect and remove infeasible (or bad) configurations from the solution space. This can be done easily with constraint propagation algorithms. It is also possible to compute the user's best configuration in terms of cost, quality or delivery time. However, with CP approach only, it is not easy to obtain high performance of maintaining the changing of configuration requirements, such configuration rules.

2.2 CP plus rule-based approach -an innovative approach

While rule-based system and CP may seem very similar from a high level, they're quite different in terms of the technologies and their applicable uses. It is commonly accepted that rule-based systems approach is not adequate for most complex configuration scenarios, which is the reason optimization engines based on CP are currently considered the state of the art. CP technology gives the end users the ability to interact with a configuration engine in the method they prefer, rather than a predetermined path developed with a business-rule modeler. However, maintenance on configuration applications considered exceptionally costly and consistently higher than expected. Rule-based systems provide features for simplifying configuration application maintenance and reducing maintenance costs. Thus, obviously, a new approach is to use constraint programming-based configuration cooperated with rules-based systems, which uses both of the two technologies together, and which allows business people to model configuration constraints and use CP technology to provide solutions.

3 CP Cooperated With Rules-Based Systems Approach

3.1 Configuration problem modeling: CP engine part

With CP plus rule-based approach, the configuration problem is represented by a set of variables, with each variable having a specified domain of possible values. The business rules that describe the regulations to be respected are formulated as constraints rules or so-called configuration rules among those variables. The CP engine applies the constraints to identify values that may lead to an invalid configuration. These algorithms are called constraint propagation algorithms. The constraint propagation algorithms are embedded into a systematic traversal of the relevant parts of the solution space, which is organized in a tree. This enables the system to automatically perform backtracking steps, implement flexible search strategies, find mathematically optimal configurations, and provide the user with explanations. Besides, the number of possible configurations may become extremely large for complex problems. As a consequence, it is necessary to

appropriately guide the search to obtain good configurations. This can be done via search strategies. Strategies control the order in which decisions, i.e., the trying of the various possible values for each field, are made. With CP plus rule-based approach, problem-specific strategies can be defined by rules:

- A user can specify the order in which the fields of a class are instantiated.
- A user can specify which objects themselves must be configured first, before the final product as a whole.

3.2 Configuration problem modeling: rule-based part

The rule-based system manages all of the mathematical and logical rules that can be applied both on the model objects and an external set of data related to the configuration (e.g. the customer profile). Thus, configuration requirements can be described as rules to state both hard and soft constraints. Hard constraints are formulated via business rules that typically belong to one of the following classes. Either rule are technical or marketing restrictions that are formulated by the supplier of the configured product or they are formulated by the customer who wants a particular part to be included in the product or the configured product to be cheaper than a given budget. Soft constraints are formulated in terms of *wishes* and *preferences*. A wish characterizes a desire that the user would like to achieve, but it may contradict some other rules. Preferences make it possible for the user to indicate that a choice in a configuration should be made over another during an automatic search. All of them are formulated in a rule-like syntax.

- A hard constraint: like regulation that must be respected
- A wish, like a rule in its structure with a major difference: if applying the wish makes it impossible to configure a solution to the problem then the engine will not apply the wish
- A preference: indicating a ranking in order of importance

3.3 Steps for application design and implementation

Unlike rule-based systems whose engine reasons directly on the application objects, an application with CP plus rule-based approach is centered on the configuration engine and must have its own object model definition and instances. Because of this, the first and most important/critical step in designing an application is the definition of the BOM (Business Object Model). This is the base of the solution build. The BOM describes the application domain objects and their relations.
The second step is to write the set of configuration rules using the BCL (Business Constraint Language). Here also, as opposed to rule-based systems, this set of rules does not express the process of finding a solution but the set of constraints the configuration engine will enforce for possible solution(s) if such exists. Thus, the design/implementation phase can be seen as iteration over these two steps. Writing rules implies modifications of the BOM. The number of iterations depends on the problem complexity and on the developer's experience with a CP plus rule-based approach.

3.4 Steps to create a configuration application

The way to develop a configuration application with the new approach can be summarized as below.

① Declare and create the object model
② State the business rules via a point-and-click editor
③ Optionally, indicate strategies to guide the search process
④ Create a configuration engine and load the object model and the business rules;
⑤ Load all the data on parts for the product from a database
⑥ Launch the search for one or several configured solutions
⑦ Optionally, use explanation mechanism to provide the end user with clear explanations on the configured solutions

In order to deploy a Web-enabled configuration application, the following additional steps may be required:

⑧ Embed the engine into an appropriate configuration server
⑨ Link the server to a Web server via the Web Connector
⑩ Create a set of Web pages on the Web server for end-user interaction (e.g., based on JSP)

3.5 Integrated Development Environment (IDE)

With CP plus rule-based approach, both problem modeling and rules should be handled. A complete set of productivity tools would be helpful, which allows developers to graphically define the product or service structure, and edit, manage and test configuration rules. The major functions of the IDE may include:

- Object Model Editor: graphical editor for easily designing the product or service structure, and editing and managing objects and relationships;
- Configuration Rule Builder: intuitive editor that enables non-technical users to edit configuration
- Application Runner: runs scenarios defined in the Configuration Rule Builder, for rapid testing;
- Search Profiler: helps tune the engine by graphically displaying the decisions made and the results obtained by the engine when searching for configurations;
- Configuration Rule Management: the Rule Builder also provides tools to store, version, query, manage and deploy configuration rules.

4. Remarks

This paper promotes and suggests a flexible, powerful constraint programming-based configuration cooperated with rules-based systems approach to solve complex configuration problems. With the constraint programming-based technology, it can obtain high performance and solving power for any configuration application. With the rules-based systems technology, it can obtain high performance of maintaining the changing of configuration requirements. ILOG JConfigurator is a pure-Java optimization software component, which uses constraint programming-based configuration cooperated with rules-based systems. With ILOG JConfigurator, all the rules can be dynamically created or changed through an interactive point-and-click editor. It also allows nonprogrammers to express configuration rules in an intuitive way. For more information, it is recommended to have a visit to http://www.ilog.co.jp/ to know more detail about it.

5. Reference

[1] Optimization Technology White Paper, ILOG, Inc. 2001.
[2] ILOG Solver, User Manual, version 5.3, 2002.
[3] ILOG JConfigurator, User Manual, version 2.1, 2002.
[4] ILOG Configurator, User Manual, version 2.3, 2002