Cincom Systems and Attar Software

Position Statement

Introduction Cincom's Business Control Systems Division (BCS) began working with Attar Software on configuration solutions in 1990. At that time, Cincom's main objective was to use Artificial Intelligence techniques to provide a product configuration component for CONTROL:Manufacturing, Cincom's ERP package. Cincom were attracted to Attar's XpertRule software because of its wide user base and the "easy to use" graphical representation of knowledge.

Initial developments centred around using native XpertRule for product configuration but this proved largely unsatisfactory due to the need to embed product and part numbers in rules based code. A significant breakthrough was in the definition of separate product option and knowledge objects linked through a configurator dictionary. This provided a configuration tool which was practical to define and maintain.

A further significant realisation achieved through research and practical implementation was that knowledge based configuration could be used to radically re-engineer the sales processes of an organisation. Configuration and associated software tools had traditionally been viewed as the domain of the engineer.The possibility of accurately configuring a product at the point of sale had largely been ignored both by industry and software solution providers. Subsequently, Cincom/Attar's developments have been geared towards providing "easy to use" configuration solutions and business systems which facilitate the re-engineering of the sales and order fulfillment processes. This has led to the development and release of CONTROL:Acquire and the key SalesConfigurator component.

Design Since 1984 Attar Software has been supplying software tools for the development of knowledge based systems. Attar's emphasis has always been on the transparency of the knowledge to the domain experts, and the ease of development and maintenance. Their XpertRule products reflects this philosophy and is widely used world-wide. In 1990, Attar Software entered into a joint venture with Cincom Systems Inc aimed at delivering configuration solutions to Cincom's clients. At that time Attar/Cincom were one of the first workers in this field to identify the limitations of rule based systems when applied to configuration applications. The decision was made to incorporate Attar's knowledge based systems technology into a new product; SalesConfigurator. The design objectives for the new product were :

- To use best of breed technologies to address the variety of problems involved in developing configuration solutions. These technologoies included knowledge based systems, rule induction, object oriented techniques, constraint based reasoning and Genetic Algorithms.20
- Ease of development and maintenance of configuration models. This is to be achieved by using a graphical development environment and graphical representation of configuration models/knowledge.

It is now broadly accepted that a product configuration solution consists of three main components; the capturing of customer requirements, the generation of the required configuration, and the reporting / exporting of the resulting configuration. The following sections explain how SalesConfigurator uses various technologies (in one integrated environment) to address each of these components.

Capturing Customer Requirements The most effective strategy for capturing customer requirements is through an interactive session in which the customer is guided into articulating his/her requirements. The customer is allowed the full flexibility of expressing his/her requirements as a set of functionalities expected from the product (environment, performance etc.) or by selecting actual product components (e.g. engine size) or a combination of both.

To capture user requirements, the developer can build graphical dialogs within SalesConfigurator or call external dialogs developed in third party tools such Visual basic or Delphi. Each graphical dialog can be linked to a number of validation (constraint) tasks. Constraint Based Reasoning is used to enforce the selection of valid combinations of product components and functionalities. The constraint logic engine is integrated seamlessly with the graphical user interface so as to restrict available selections based on selections already made on the current or previous graphical dialog. Note that this represents a pro-active method of imposing constraints which dynamically alters the available selections on a dialog to prevent the user from making invalid selections. The user can freely alter selections on sections of the dialog to see the restriction that imposes on other sections. In contrast, reactive constraint validation simply reports back on invalid combinations of selections.

SalesConfigurator allows constraint logic to be expressed graphically as a set of validation trees as illustrated in the following tree for validating combinations of options 1 to 3: (note that the following is a textual representation of the graphical trees displayed in Sales Configurator).

```
Option1
      |--------- A : Option2
      |                      |------- X : Option3
      |                      |                    |------- 1 : INVALID
      |                      |                    '------- 2: VALID
      |                      '------- Y : Option3
      |                                           |------- 1 : VALID
      |                                           '------- 2: VALID
      '--------- B : Option2
                            |------- X : Option3
                            |                    |------- 1 : VALID
                            |                    '------- 2: INVALID
                            '------- Y : Option3
                                                 |------- 1 : VALID
                                                 '------- 2: INVALID
```

The attributes appearing in validation trees can be functional requirements or specific components). Validation trees can be developed manually or, alternatively, they can be automatically generated (induced) from validation tables of the following format :

Option1	Option2	Option3	Status
*	*	*	VALID
A	X	1	INVALID
B	X	2	INVALID
B	Y	2	INVALID

The above table expresses INVALID exceptions. Validation tables can also be expressed as VALID exceptions.

Linking constraint logic to event driven graphical dialogs requires a fast constraint engine. Executing complex constraint (validation) trees can be orders of magnitude faster than executing validation tables (rules). We therefore believe that the approach of induction of validation trees from tables results in ease of maintenance and speed of execution.

Case Based Reasoning is used in complex configurations to assist the customer in identifying the generic product templates that best fit his high level specification of requirements. Rule induction is used (at development time) to group generic configurations into templates based on high level attributes. These templates are then adapted to the specific customer requirements.

Configuration Engine & Model The configuration engine transforms the captured user requirements into a product configuration using the configuration model of the product. The technologies used for product modelling are rule based and object oriented techniques.

Object oriented techniques are used to model the components of the product and the generic components hierarchies (assemblies). Components hierarchies are either defined staticly at development time, dynamically at runtime or combination of both. The detailed components data are stored in a product data base. Generic hierarachies are represented as graphical trees.

The inference strategy of the configuration engine is directed by a 'main agenda' which controls inference from the various components and subassemblies of the model. Each component or subassembly in the product model has an agenda (method) which invokes the selection rules and calculations for that component. Component selection rules are expressed as decision trees with functional requirements and other components being used as selection attributes. The following is an example selection tree for a component using two functional requirements and another component as attributes.

```
Function1
      |--------- A : Function2
      |                     |------- X : Component3
      |                     |                   |-------  selected:NOT SELECTED
      |                     |                   '------- 2:  SELECTED
      |                     '------- Y : SELECTED
      '--------- B : Function2
                           |------- X : NOT SELECTED
                           '------- Y : SELECTED
```

Selection trees, in a similar way to validation trees, can be developed manually or induced automatically from selection tables again resulting in transparency and ease of maintenance of the knowledge.

Resource Optimisation: In many configuration applications there is a need to apply spatial constraints, typically minimising the area required to fit the configured components or the number of components (e.g. cabinets) required to house the configured product. Sales Configurator offers two types of technologies for addressing this requirement. Rule based resource allocation is used if the allocation task is simple and does not require complex iterations. For complex tasks, Sales Configurator offers a Genetic Algorithm Optimisation Engine.

Reporting / Exporting Configurations Once a configuration session is successfully completed, there are a number of ways of reporting the resulting configuration:

- show the user a graphical representation of the product
- generate a quotation or proposal
- interface to a CAD system
- interface to a Sales Order Processing system
- interface to an MRP system

Whilst SalesConfigurator supports the development of graphical reports, its strength lies in its ability to process the generated product configuration and to interface to word processors, CAD, MRP or Sales Order Processing software packages in order to produce the required report/export. This is achieved through the procedural processing capabilities of SalesConfigurator which allows it to use DDE and OLE2 servers and interface to DLLs, ODBC data sources and APIs.

Concluding Remarks Cincom & Attar have delivered configuration solutions to a large number of organisations in the Engineer-to-order and assemble-to-order environments. These clients are from various manufacturing sectors including telecom, power engineering, specialist vehicles, machinery manufacturers and computers. For many applications, SalesConfigurator has enabled organisations to develop configuration models within weeks. Furthermore, in most cases, these models are maintainable by computer literate product engineers and sales/marketing staff.

We believe that it is all too easy to develop a configuration application that appears to be full of 'bells and whistles' when it is run, whilst hiding the complexity of the configuration model underneath. The real challange for a product configurator is to appear full of 'bells and whistles' at the model development and maintenance stage without requiring an Artificial Intelligence specialist. We firmly believe that SalesConfigurator meets this challenge.