Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1998), *12*, 295–300. Printed in the USA.

POSITION PAPER

Configuration research and commercial solutions

DAVID W. FRANKE
Trilogy Development Group, Inc., 6034 W. Courtyard Dr., Austin, TX 78730, U.S.A.

(RECEIVED October 31, 1997; ACCEPTED February 5, 1998)

Abstract

In this paper we intend to motivate various research areas in configuration, based on our experience in developing commercial configuration solutions. Informal definitions are given for the configuration task and for configuration specification and description languages. We also offer an abstract characterization of the issue addressed in configuration models, most often represented as constraints or rules. With these definitions and examples of configuration problems from various domains, we motivate research in (1) a common ontology for configuration, (2) function representation and functional reasoning in configuration, and (3) scaling configuration to large problems.

Keywords: Configuration; Abstract Characterization; Commercial Configuration Solutions

1. INTRODUCTION

In developing commercial configuration solutions, we have developed and worked with all aspects of software tools required for developing and deploying configuration solutions. These software tools include configuration "engines" (that perform the configuration task), configuration modelling languages, configuration models (that represent domain and problem specific knowledge), user interfaces for configuration model builders and end users, and deployment infrastructure for configuration applications and configuration models and data. These commercial configuration solutions have been developed for many problem domains, including computers (PC, workstation, internetworking, mainframe, and supercomputer hardware, software, and services); telecommunications systems (e.g., central office switches, PBXs, and internetworking); transportation (automobiles, trucks, and airplanes); modular furniture; custom manufactured materials (e.g., steel); industrial products (e.g., valves, actuators, and controls); medical systems and services.

In light of this experience,[1] we offer some observations regarding research directions. In particular, we will discuss the potential impact of advances in

- Definition of ontological alternatives for configuration,
- Function representation and functional reasoning for configuration, and
- Scaling representation techniques and solution techniques for large problems.

In the next section, as a preface to the discussion of research directions, we provide (1) definitions of our terminology, (2) a perspective on the expectations of organizations that want to provide automated configuration solutions, and (3) a perspective on the expectations of end users of these configuration applications.

Reprint requests to: David W. Franke, Trilogy Development Group, Inc., 6034 W. Courtyard Dr., Austin, TX 78730, U.S.A.

[1]The perspective and opinion offered here is based on our experience in developing commercial configuration solutions, with specific focus on solutions that can be deployed to end users and that provide a significant return on the customer's investment. While this information is relevant to research activities, it is not presented as a research result, and our focus has not been fundamental research but rather commercial configuration product and solution development.

2. THE CONFIGURATION TASK

We give a definition of the configuration task that elaborates the definition in (Mittal & Frayman, 1989). We define in more detail characteristics of the input specification and we give an abstract characterization of the types of conditions represented in configuration models. The configuration task is defined in the straightforward way: produce a configuration from an input specification of the desired configuration. The input specification is assumed to be incomplete, either because it specifies a subset of the features or content of the desired final configuration or because the specification is an abstract versus detailed definition of the desired final configuration.[2]

2.1. Requirements specification

The input specification language must allow expression of the requirements as understood by the user. For example:

- *Products and options*—automobile models (coupe, sedan), options (sunroof, 5-speed).
- *Individual components (parts)*—200-MHz processor, four 64-Mbyte SIMMs.
- *Structure*—room dimensions, explicit positions for modular office furniture.
- *Function*—truck that transports 20,000 lbs. over road grades up to 6% grade such that at speeds between 90 and 100 KPH no shifting is required.
- *Preferences*—price sensitivity with respect to performance.

Input specifications described in these terms can be proposed in different ways:

- *Mixed specification*—includes specific components, structure, and function.
- *Upgrade specification*—an existing configuration and a new specification in terms of any combination of product, option, specific component, structure, and function.
- *Template specification*—partially configured starting point.
- *Proposed configuration*—presumed to be correct, submitted for verification.

In all forms of input specification, it is assumed that the specification can be entered incrementally, with the implications of each individual input considered and reflected in the user interface to guide the user to appropriate subsequent inputs. Implications of previous choices can be manifested as:

- Valid, required, or unavailable (invalid) selections.
- Valid quantities and ranges.
- Recommended or default selections.
- Recommended or default quantities.
- Explanations—Why is this choice unavailable? Why is this choice included?
- Verification—What was incorrect? Recommended changes (add/delete/replace).

2.2. Configuration description

The configuration generated by the configuration task is described in terms of content and structure. Configuration content is described in terms of physical parts or components (e.g., power supplies and cables), abstract parts (e.g., services and information), and functions in the configuration. Configuration structure is described in terms of physical placement, such as position in a room or in a cabinet described in some coordinate system, and in terms of topological relationships. The topological relationships may be hierarchical, such as a controller card in a slot of a card rack in a cabinet, or nonhierarchical, such as storage device is connected to controller via some cable or composed connection path.[3]

2.3. Configuration content issues

Given this definition of the configuration specification language and the product of the configuration task, the issues that must be resolved in the configuration task with respect to configuration content are:

- *Compatibility*—can the components be used together?
- *Completeness*—are all components required to meet the input specification included?
- *Capacities*—have all internal and external capacity requirements been met?
- *Optimality*—is the configuration optimal with respect to some measurement criteria?

These issues, compatibility, completeness, capacity and optimality, must be considered in terms of the structure of the configuration. For example,

- *Compatibility:* The topological relationship between components can influence their compatibility conditions. Storage devices that cannot be connected to the same controller, but otherwise can occur simultaneously in the configuration.
- *Completeness:* Ballast is required when heavy components are placed high in a cabinet. Cross supports are

[2] It may not always be the case that the input specification is incomplete. Verification of proposed configurations generated by other means is often required in commercial uses of configuration tools, and input for this verification step will be a (presumably) complete specification.

[3] A configuration specification language should also be able to express the result of the configuration task, namely a final configuration. This is important when upgrading an existing configuration, because the existing configuration is part of the input specification.

required for long, linear runs of modular furniture panels.

- *Capacity:* The physical placement of power supplies in cabinets and the physical placement of components that consume power in cabinets is critical to determining whether the power requirements of the components in any individual cabinet exceed the power provided in any individual cabinet.
- *Optimal:* The physical placement of components in discrete locations (e.g., cabinets or rooms) should be such that additional functionality (upgrades) can be added in the future in any of the discrete locations with minimal relocation of existing components.

2.4. Configuration structure issues

With respect to configuration structure, the issues of structure to consider are:

- *Physical placement*—location in some space (coordinate system)
- *Topology*—hierarchical and nonhierarchical (connectivity)

Large telecommunications product configurations, multicabinet computer configurations, and internetworking product configurations incorporate all these aspects of structure in their configuration specification, description, and solution.

2.4.1. Physical placement

Configurations containing multiple cabinets must position each cabinet in a user-specified site (room, multiple rooms, multiple floors or a building, or even multiple buildings). Considering the simplest case, a single room, a floor plan specifying the position of each cabinet is part of the configuration. Cable routing is also required in these configurations, requiring the determination of the specific path through the space that the cable occupies.

2.4.2. Topology, hierarchical

Common hierarchical topology in these configurations is function specific components (e.g., storage devices, circuit packs, controllers) are placed in mounting racks that are in turn placed in cabinets. For multiroom or multibuilding sites, this hierarchy extends to rooms within floors within buildings within campuses.

2.4.3. Topology, nonhierarchical

These configurations require the generation of logical connections between components or specific ports of component. Beyond the considerations of the physical routing of cables, logical connections must be represented and reasoned about.

2.5. Multiple views

When entering the input specification, or when examining the results of the configuration task, there are different views the user may want to use. The common views for the input specification and the resulting configuration are:

- *Function or requirements view*—input specification or configuration result in terms of user requirements or functional requirements.
- *Structural view*—structural relationships given in the input specification or generated in the configuration result.
- *Quote view*—components of the input specification or the configuration result organized as products that have associated prices.
- *BOM (Bill-of-Materials) view*—components of the input specification or the configuration result, presented as a flat list or a presented as a hierarchical structure reflecting product structure defined for marketing or manufacturing purposes.

The ability to move between different views of the input specification or the generated configuration is required. When moving among different views, the current state of the input specification or the configuration result must be reflected in the current view.

3. RESEARCH ISSUES

3.1. Ontology

Whether discussing configuration problems and solutions with prospective customers or reading technical articles on configuration, the terminology used in configuration specification languages and configuration description languages varies, as does the terminology used in describing configuration solution techniques. Terms such as "class," "component," or "property" are used with different semantics. This is particularly prevalent in configuration solutions, manual or automated, developed inside organizations for internal use. Such solutions use existing terminology of the organization, describing products and associated configuration issues in engineering, manufacturing, marketing, or sales contexts. Using existing terminology makes the solution understandable by the organization, and hence sellable within the organization, an important consideration when promoting an internal project. These solutions are also proprietary, and consequently little or no information regarding the solution is exchanged with the external community. The research community provides a more homogeneous terminology via the comparison and cross-referencing of technical articles.

Terminological confusion is sometimes a result of ontological confusion. To understand existing or proposed configuration solutions, exposing and examining the ontological choices of the configuration specification (input) language, the configuration description language, configuration model representation language, and the solution technique is key. Once the ontological choices are understood, it is easier to resolve terminological differences, and consequently to evaluate the capabilities of individual approaches and to compare alternative approaches.

A valuable research activity is to enumerate the existing ontological choices made for

- Configuration specification (user requirement).
- Configuration result (generated configuration) description.
- Configuration model.
- Configuration solution techniques.

Our experience is that ontological elements, once identified, are found in most if not all language used to describe the aspects of configuration solutions listed above. All configuration solutions we are aware of use a component or part ontology because the output of the configuration task is described in terms of the components that make up the configuration. Other common ontological elements are function, component-to-component relationships, and component-to-function relationships. Surveys of configuration techniques that address some ontological issues but do not focus on ontology per se are (Balkany et al., 1993; Stefik, 1995; Darr & Dym, 1998). This special issue contains a paper that surveys the ontological choices of various configuration solutions and attempts to unify some of the terminology.

3.2. Functional reasoning

Function, as an ontological element, is a means for expressing input requirements in the language of interest to the user, namely their needs. Combining function representation capability with functional reasoning is one approach to adding needs analysis to a configuration solution. A configuration solution that performs "needs analysis" accepts an input specification defined in terms of functionality required by the user (i.e., the needs), and from that specification generates a configuration that provides the required functionality. To achieve this mapping from function to a configuration of components that realize the function, a configuration solution must provide:

- Explicit representation of function.
- Function-to-component mapping.
- Techniques for reasoning about function.

Consider a customer request for a computer system that can be used as a database transaction server, with the additional requirements that (1) the computer system support 1000 transactions (of some nominal size) per minute, and (2) the computer system must be expandable (i.e., the customer should be able to purchase additional components incrementally as their processing demand increases) to support up to 5000 transactions per minute. To express this requirement, the input specification language must be able to represent and quantify "database transaction" and be "expandable." Further, the configuration technique must provide a means for reasoning about these functions, and the configuration description language must be able to characterize the resulting configuration in terms of database transactions supported and future expansion capabilities.

The MAPLE system (Bowen, 1986) uses a functional representation at the level of connecting components. Resources as an ontological element of configuration is described for the COSMOS system (Heinrich & Jungst, 1991). We provide a function representation in our configuration solution, specifically in terms of resource provision and resource consumption (Downing, 1990). While the notion of function as resource, with resource providers and consumers has proven to be quite useful as a function representation as well as an abstraction mechanism in configuration models, this is only one approach to function representation and reasoning. Leitgeb and Pernler (1996) and Lee et al. (1992) have investigated functional representation and reasoning in configuration. The functional reasoning community [cf. Chandrasekaran (1994), and Umeda and Tomiyama (1997)] has developed a broad range of function representation and reasoning techniques, and further research in applying existing functional representation and reasoning techniques to configuration and in investigating new functional representation and reasoning techniques holds the promise of improvements to configuration solutions.

3.3. Scaling to large problems

Given the inherent complexity of solving configuration problems (Frayman & Mittal, 1989), large problems present a representation and computation challenge to configuration solutions. Problem complexity and problem size manifest themselves in configuration in multiple ways.

- *Large numbers of possible components from which to choose*—the configuration model and associated component database we have constructed for configuring products in the personal computer (PC) industry has over 20,000 unique components, and continues to grow daily. While configured PCs are straightforward to describe when completed (i.e., small), the space of possible component selections and hence possible configurations is large.
- *Large numbers of instantiated components in the final configuration*—we have constructed models with our customers and have configured large telecommunications switches that contain more than 20,000 instantiated components. While the configuration model may contain only tens or hundreds of unique components, the final configuration can be quite large in that individual components may be instantiated hundreds or thousands of times. Multiple instantiations are required when instance specific data (e.g., placement and connectivity) is required.

- *Large numbers or complex configuration conditions*[4]—configuration of complex products where all of the components are manufactured by a single vendor can have this characteristic. An example of such a product is a supercomputer. In contrast to product categories where many of the components are commodities that can be purchased from many vendors (e.g., personal computers), intercomponent interfaces have not been standardized, but rather have highly customized and specialized interfaces and configuration conditions. Configuration conditions can outnumber the components by an order of magnitude, because few generalizations can be made in the configuration conditions, and individual combinations of components often require distinct configuration conditions.

In addition to increasing the space of possible combinations, the desire to identify an optimal configuration increases the computational requirement further. For example, optimal configuration of PBX's (Private Branch Exchanges, a category of telecommunications equipment) is critical. The PBX market is highly competitive, and optimal configurations can give a PBX vendor cost advantages over their competitors, leading to a lower offer price and ultimately to winning the customer's business.

As the complexity of the configuration problem increases, the value delivered by a configuration solution increases accordingly. Obtaining a correct configuration eliminates errors for the organization offering the product, thereby making the organization's sales and order management processes more efficient. Correct configurations improve customer satisfaction in that the customer receives a product customized to their needs. With respect to scaling, significant value is realized when configurations can be generated in a timely fashion. Configuration systems that (1) produce good or optimal solutions quickly and (2) utilize minimal space (memory and disk space) provide significant value in that they allow exploration of different customer requirements and evaluation of alternative solutions on a computational platform accessible to those involved in the sales process. These configuration systems also allow organizations to quickly respond to changes in customer requirements during the course of the sales cycle, during order processing, or during manufacture of the desired product.

Problem decomposition can be used to limit the space of possible solutions under consideration. This approach is being explored currently in research (Fleischanderl & Haselboeck, 1996; Weigel & Faltings, 1996; Heinrich & Jungst, 1996). In addition to problem decomposition, Darr and Birmingham (1996) also address run time efficiency by distributing the computation. We provide primitives for problem decomposition in our configuration solution, and this technique has proven useful in addressing large configuration problems, network configuration problems, and problems where independently developed configuration models must be integrated to construct the configuration specified in the input requirements. Many complex problems, however, cannot be decomposed into independent or nearly independent subproblems. Further research into problem decomposition applied to configuration problems, and for techniques for nondecomposable problems is needed.

[4] We use the term "condition" to avoid any assumptions implicit in rule-based, constraint-based, or other representation.

4. SUMMARY

There are many research topics, not configuration specific, which will have an impact on the application and use of configuration solutions. Business transactions and processes are moving to the Internet, and configuration is one of many applications being applied to these transactions and processes. Any research that addresses application deployment, distributed processing, or electronic commerce on the World-Wide-Web will improve the applicability, value, and importance of configuration solutions in this space.

The research topics listed in this paper is not a comprehensive list of relevant research areas for configuration, but rather a subset of those research topics whose results can have a near term impact on configuration solutions applied to real world problems.

ACKNOWLEDGMENTS

The ideas and examples described here have been developed through the cooperative effort of many individuals at Trilogy Development Group, and the author gratefully acknowledges the contributions of all those individuals. I would also like to thank the reviewers for their comments.

REFERENCES

Balkany, A., Birmingham, W.P., & Tommelein, I.D. (1993). An analysis of several configuration design systems. *AIEDAM 7(1)*, 1–17.

Bowen, J. (1986). Automated configuration using a functional reasoning approach. *Artif. Intell. Applications* 79–106.

Chandrasekaran, B. (1994). Functional reasoning: A brief historical perspective. *Applied Artificial Intelligence*.

Darr, T.P., & Birmingham, W.P. (1996). An agent architecture and algorithm for solving distributed configuration-design problems. Report No. FS-96-03.

Darr, T.P., & Dym, C.L. (1988). Configuration design: An overview. In *The Handbook of Applied Expert Systems*, (Liebowitz, J., Ed.), pp. 21.1–21.15. CRC Press, Boca Raton, Florida.

Downing, K. (1990). The qualitative criticism of circulatory explanations via bipartite teleological analysis. *Proc. 1990 Workshop on Qualitative Reasoning*.

Fleischanderl, G., & Haselboeck, A. (1996). Thoughts on partitioning large-scale configuration problems. Report No. FS-96-03.

Frayman, F., & Mittal, S. (1987). COSSACK: A constraint-based expert system for configuration tasks. *Proc. Second Int Conf. Applications of AI to Engin.*

Heinrich, M., & Jungst, E.W. (1991). A resource-based paradigm for the configuring of technical systems from modular components. *Proc. Seventh IEEE Conf. AI Applications (CAIA'91)*, 257–264.

Heinrich, M., & Jungst, E.W. (1996). The resource-based paradigm: Configuring technical systems from modular components. Report No. FS-96-03.

Lee, C.L., Iyenger, G., & Kota, S. (1992). Automated configuration design of hydraulic systems. In *Artificial Intelligence in Design '92*, (Gero, J.S., Ed.), pp. 61–82. Kluwer Academic Publishers, New York.

Leitgeb, M., & Pernler, S. (1996). Functional and structural reasoning in configuration tasks. Report No. FS-96-03.

Mittal, S., & Frayman, F. (1989). Towards a generic model of configuration tasks. *Proc. Eleventh Int. Joint Conf. on Artificial Intelligence* (IJCAI-89), pp. 1395–1401. Morgan Kaufmann, San Mateo, California.

Stefik, M. (1995). Configuration. In *Introduction to Knowledge Systems*, pp. 608–669. Morgan Kaufmann Publishers, San Mateo, California.

Umeda, Y., & Tomiyama, T. (1997). Functional reasoning in design. *IEEE Expert/Intelligent Systems & Their Applications 12*(2).

Weigel, R., & Faltings, B. (1996). Abstraction techniques for configuration systems. Report No. FS-96-03.