



A Database Administration Tool to Model the Configuration Projects

Shafiee, S.; Friis, S. C. ; Lis, L. ; Harlou, U.; Wautelet, Y. ; Hvam, L.

Published in:

Proceedings of 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)

Link to article, DOI:

[10.1109/IEEM.2018.8607654](https://doi.org/10.1109/IEEM.2018.8607654)

Publication date:

2019

Document Version

Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):

Shafiee, S., Friis, S. C., Lis, L., Harlou, U., Wautelet, Y., & Hvam, L. (2019). A Database Administration Tool to Model the Configuration Projects. In *Proceedings of 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 341-345). IEEE.
<https://doi.org/10.1109/IEEM.2018.8607654>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Database Administration Tool to Model the Configuration Projects

S. Shafiee¹, S.C. Friis², L. Lis², U. Harlou², Y. Wautelet³, L. Hvam⁴

¹Department of Mechanical Engineering, Technical University of Denmark, Lyngby, Denmark

²Center for Product Customization, Lyngby, Denmark

³Centre for Information Management, KU Leuven, Warmoesberg, 26, 1000 Brussels, Belgium

⁴Department of Management Engineering, Technical University of Denmark, Lyngby, Denmark
(sashaf@dtu.dk, scf@pfmp.com, lli@pfmp.com, uha@pfmp.com, yves.wautelet@kuleuven.be, lahv@dtu.dk)

Abstract - Up-to-date product knowledge modelling, documentation, and validation for the configuration projects is crucial for maintenance, further development, system quality and communication with domain experts. Product models are the main communication and documentation tools used in configuration projects. Furthermore, interest in the adoption of CASE tools has escalated because of the important role they play in supporting the software development and documentation process. Moreover, studies show that CASE tools have a positive impact on quality and productivity in IT projects. Moreover, due to the reported amount of the complexity in configuration project and the needed time and resources, CASE tools can be applied to solve the challenges of development, documentation and communication. This paper aims to demonstrate the possibility of developing a CASE tool for configurators and apply the CASE tool in one case example at a case company.

Keywords – Configuration, CASE tool, development, documentation, communication

I. INTRODUCTION

Configurators have entered a new stage of maturity and have recently received increased attention from both researchers and practitioners. Configurators support decision-making processes in the sales and engineering phases of a product, which can determine the most important decisions regarding product features and cost [1]. Widely used in various industries, configurators can bring substantial benefits, such as shorter lead times for generating quotations, fewer errors, increased ability to meet customers' requirements regarding product functionality, the use of fewer resources, optimized product designs, less routine work and improved on-time delivery [2]–[7].

There are several challenges reported from configurators which can be categorized based on [6]. Categories of challenges related to implementation and utilization of configurators include: IT related, Product modeling, Resource constraints, Organizational, Product-related, and Knowledge acquisition.

A database management system (DBMS) is a collection of programs that manages the database structure and controls access to the data stored in the database [8]. In a sense, a database resembles a very well-organized electronic filing cabinet in which powerful software, known as a database management system, helps manage the cabinet's contents. There are several benefits using DBMS such as improved data sharing, improved data security, better data integration, data consistency,

improved data access, improve decision making, and increase end-user productivity [8]. Multiple IT tools have been developed to assist with data management in IT and configuration projects. For example, the Computer Aided Software Engineering (CASE) tool applies a set of tools and methods, generating a software system to support the desired end results: i.e. a defect-free and maintainable software product [9]. The CASE tool can be used for software documentation; however, CASE tools are not specifically designed for configuration, but rather for the general development of IT projects.

Previous researchers address different challenges and propose a specific solution for several of the mentioned challenges such as documentation, knowledge acquisition, resource constraints and product modelling challenges [6], [7], [10]. As such, there is still a critical need in the industry for an efficient automatic solution to be found for documenting the knowledge of configurator, communicating with domain experts and validating product knowledge; while, the automated transferring of knowledge to the configurator is possible. Recently, Shafiee et al. [10] propose a framework to document the knowledge inside the configurator and extract them to be able to communicate and maintain the knowledge.

This study have a step further to introduce a framework which document and maintain the knowledge in a user friendly system outside the configurator and automatically model them inside the configurator. This will also provide the possibility of maintaining the knowledge in only one place and out of configuration system and also will remove all the manual work. In accordance with the focus of the study, the following research questions have been developed:

1) How can we document and model the product information (attributes and rules) outside the configurators?

2) How can all the product information (attributes and rules), modelled and documented out of the configurator, be transferred to the configurator without any need for further IT development and modelling?

II. RESEARCH DESIGN

In order to develop our framework, first, we studied the available literature on modelling methods, documentation methods, and CASE tools and their benefits. To provide a foundation for the proposed framework, we studied why and how CASE tools are of benefit. Based on evidence from the literature and our

experiences of working with multiple commercial configurators, we recognized that most of commercial configurators required the same knowledge to generate the solution space, which should be included in the CASE tool as well. After further investigation, we developed an IT solution as the CASE tool to contain the knowledge and get connected to the configurator to generate the product configuration model. This CASE tool developed in a short time and without significant investment.

We tested the proposed framework and the IT CASE tool in order to discover: (1) whether the structure, attributes and constraints of a commercial configurator could be documented and maintained in the CASE tool to full extent, (2) how to link the CASE tool with the configurator and (3) whether the generated configurator would perform completely without errors at the case company.

III. LITERATURE STUDY

A. Modelling and documentation of the product knowledge

Product modelling is a method of representing the structure and knowledge of the product on a relatively visual, abstract level to ensure that they are understandable to all persons concerned. Configuration models specify the set of possible configurations (solutions) and a configuration model together with a defined set of (customer) requirements are the major elements of a configuration task (problem) [11]. Four basic representations for modelling product families for configurators are shown in Fig. 1 [12].

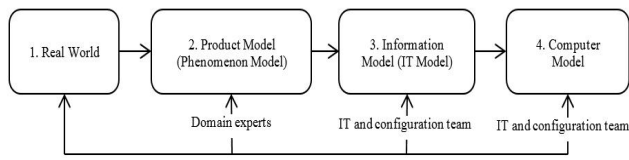


Fig. 1. Four basic representations of product modelling for configurators, revised from Duffy et al. [12]

Modelling tools are used for the communication and documentation of configuration projects. Product modelling is used to handle the growing complexities of software development, enabling engineers to work and communicate at higher levels of abstraction, and have a proper documentation of both product and project knowledge [13]. In addition, these techniques increase the ability to share knowledge between units, which can contribute significantly to an organization's performance [14]. Different modelling techniques are available for configuration projects [13], [15]–[17]. A detailed analysis of the product range is necessary – the use of the Product Variant Master (PVM) together with CRC cards is suggested for this purpose (Fig. 2), both being based on UML techniques.

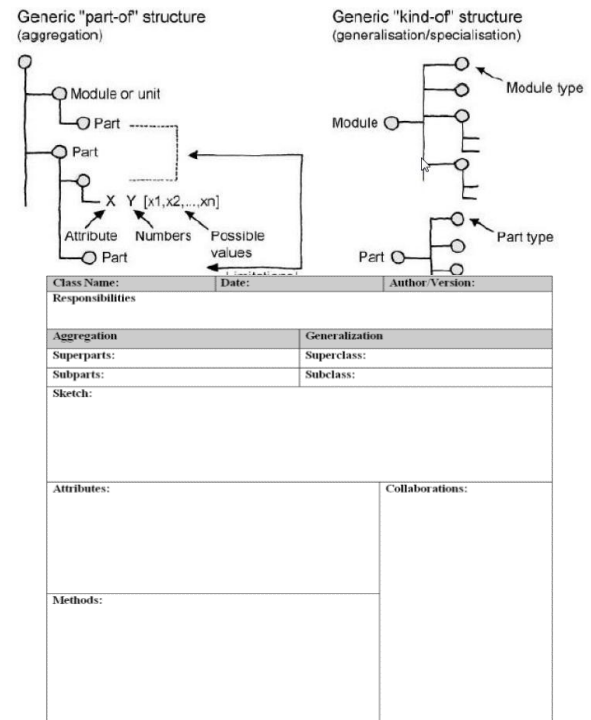


Fig. 2. Structure of the PVM and CRC cards [1].

One of the main challenges with configurators is the difficulties in documentation, maintenance, and communication, which can lead to incomplete and outdated systems that are difficult to understand [18][19]. Documentation is a vital part of all IT projects, as it is used for sharing knowledge between people and reducing knowledge loss, when team members become inaccessible. For a company using a configurator, it is therefore crucial to have an efficient system for documenting the structure, attributes, and constraints modelled within the system, as well as to facilitate communication between configurators developers and domain experts¹. The documentation of configurator includes modelling, maintaining and updating the product model, and storing all information related to the products' attributes, constraints and rules inside the configurator [1].

B. Computer-aided software engineering (CASE) tools

A CASE tool provides an automated framework for the Systems Development Life Cycle (SDLC). CASE uses structured methodologies and powerful graphical interfaces. Because they automate many tedious system design and implementation activities, CASE tools play an increasingly important role in information systems development [8].

There are various benefits from CASE tools can be listed as:

1) *Document Handling* [20] [8]: The most obvious area for tool support is document handling. Traditional inspection requires the distribution of multiple copies of

¹ The experts who provide domain knowledge of the process of performing the task and the data content, as well as quality assurance, verification support [22].

each document required. Apart from the cost and environmental factors associated with such large amounts of paper, cross-referencing from one document to another can be very difficult.

2) *Individual Preparation [20]*: There are various levels of integration, from simply reporting defects to actually producing an annotation relating to the defect for the reviewer to examine.

3) *Data Collection [20]*: Computer support allows metrics from the inspection to be automatically gathered for analysis. This removes the burden of these dull but necessary tasks from the inspectors themselves, allowing them to concentrate on the real work of finding defects.

4) *Easier maintenance of application systems [8]*: The documentation and ease of access to them make the maintenance more efficient. Better communication and overview for experts and project members are provided.

5) *Standardization of systems development methodologies [8]*: One of the CASE tools' most important components is an extensive data dictionary, which keeps track of all objects created by the systems designer.

6) *Data dictionary [8]*: The CASE data dictionary stores data flow diagrams, structure charts, descriptions of all external and internal entities, data stores, data items, report formats, and screen formats. A CASE data dictionary also describes the relationships among the components of the system.

7) *Quality of communication [8]*: A CASE environment tends to improve the extent and quality of communication among the developers, the application designers, and the end users.

8) *Cost and budget management [8][21]*: CASE tools are aimed to address the difficulties of developing high quality, complex software on time and to budget.

IV. FRAMEWORK DEVELOPMENT

The IT tool developed in this study needs to exhibit certain characteristics to produce valid results that will be

of value to academia as well as industry. these factors could be mentioned as: support for a large part of the SDLC, flexibility, intelligence in the tool, a professional CASE tool useful for industry, and support for the Object-Oriented paradigm that industry is using on new projects [21].

Based on the talks and discussions with the stakeholders, the CASE tool has been developed according the provided categorization (Fig. 3). These categorizations fulfill the required data based on PVM and CRC Cards. The categorization is also equivalent to the needed knowledge and structure inside the commercial configurators.

The framework suggested for developing this CASE tool is illustrated in Fig. 3. First, the initial product model is built using different user interface in the CASE tool. The CASE tool is available to all the stakeholders and different stakeholders can provide the knowledge. Relevant stakeholders, and based on the responsibilities, can document, maintain and update the product knowledge inside CASE tool. This tool will be the database of the product. For future versions of the configurator, updates will be made from within the CASE tool and the updated product models will be generated directly from the CASE tool to the configurator. The model generation to the configurator is automatic and this will reduce the needed time and resources.

V. CASE STUDY RESULTS AND IT TOOL DEVELOPMENT

The IT tool is developed in Microsoft Visio and Microsoft Excel as a diagramming and vector graphics application. The whole development of the IT tool is divided in two sections. First, we manage to provide a friendly IT environment with different categories to let the users insert the knowledge in relevant pages. The second part of development devoted to link the IT tool to our commercial configurator to be able to transfer the knowledge. Finally we will discuss the whole approach.

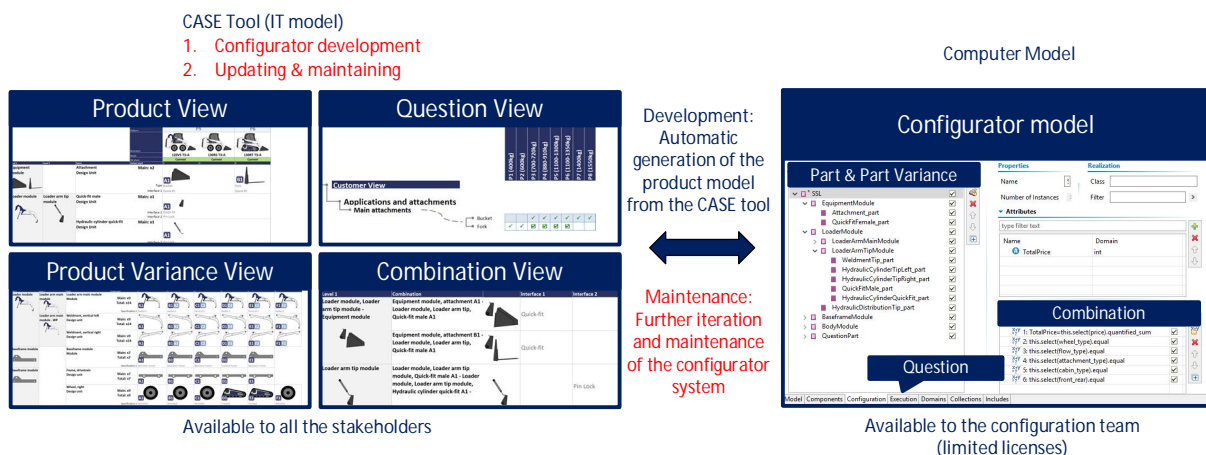


Fig. 3. The process of automatically generation of the configurator from the CASE tool

Summarizing, there are various benefits to be mentioned as the result of this automation and few of them are mentioned below:

- 1) Similar part structure through the configurator
- 2) Transparent configurator data

- 3) Improve maintenance ability
- 4) Ability to delegate data responsibility
- 5) Improved documentation ability
- 6) Support configurator development phase

Fig. 4 demonstrates the architecture design of how the data has been transferred to the configurator.

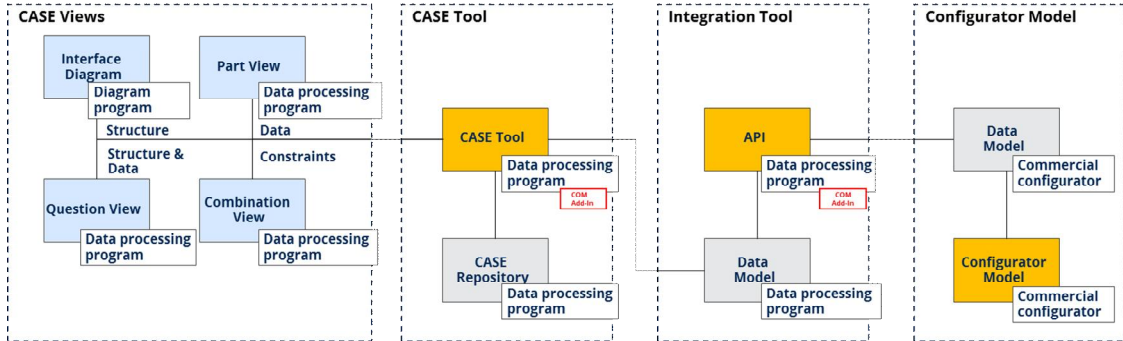


Fig. 4. Configurator architecture diagram to demonstrate how the different modules of the configurator connects and which type of programs the different part of the configurator is using.

A. Interface design

The interface diagram is used to define the configurator structure based on modules and interfaces. The module can be functional units that all together describe all the functions in the product configurator. The interactions between the modules are defined as interfaces. Interfaces are a way to manage the data type between the modules (Fig. 5).

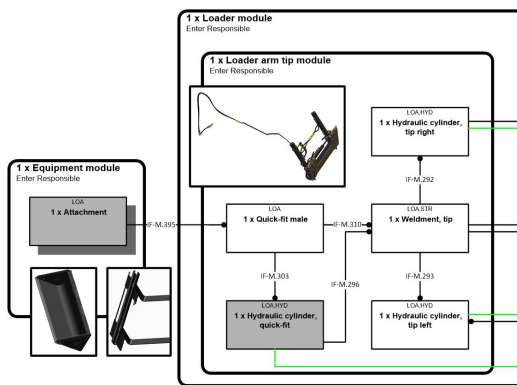


Fig. 5. Snippet of an interface diagram to demonstrate an attachment part in the equipment module interfacing to a quick-fit male part in the loader arm tip module

The interface diagram is used to define the configurator structure based on modules and interfaces. The module can be functional units that all together describe all the functions inside the configurator. The interactions between the modules are defined as interfaces.

We develop an IT CASE tool to be able to manage all the knowledge instead of control everything in the configurator. This way we are able to document, manage, and maintain the configurator knowledge through the data model (Configuration logic). So, we studied the data needed inside the configurators to define the solution space and based on that designed our CASE tool.

Therefore, the data modeled before transferring to the configurator are divided into the different types of data including:

- 1) Product data
 - o Product information (Only control product information: Component dimensions, engineering hours, colors etc.)
- 2) Combination data
 - o Component combination (different ways that the components can be combined)
 - o Constraint solution space (the feasibility and allowance for combining the components)
 - o Component & question part combination
- 3) Question data
 - o Guiding domain experts' information (the knowledge provided for product introduction and overview)

This division provide a nice user interface with the borders to categorize and cluster all the required knowledge in our IT CASE tool. There will be no need for major training or required IT expertise to manage the knowledge in this IT system. The user interface of the IT system is illustrated in Fig. 6. The developed CASE tool include multiple categorizes as below:

- 1) The product view: It facilitates the understanding, explaining and also validating the product knowledge in the section called product view which help to manage the whole product portfolio.
- 2) The part variants view: It facilitates the understanding, explaining and also validating the products' detailed parts and components knowledge in the previous section called product view.
- 3) The combination view: The combination view is used to constrain the full potential solution space. The combination view therefore show the actual solution space by showing which modules that can be combined.

4) The question view: The question view is design to facilitate and help the user to manage the question structure and rules needed inside the configurator. This section creates the question structure and data to setup the correct questions that will guide the user to the solution to meet his requirements.

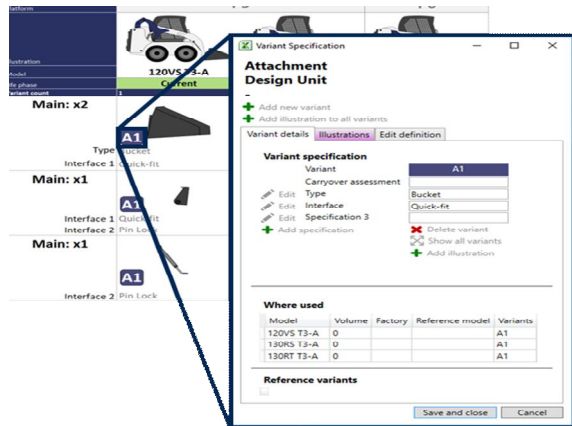


Fig. 6. User interface of the IT system and the product specification window for editing

To make our IT tool to perform as a CASE tool, we manage to transfer the data automatically to our configurator. For this study, the available commercial configurator were evaluated. The transfer of knowledge from IT tool to configurator is possible using Microsoft Excel and available default Excel plugin for the commercial configurator used in this specific case study.

VI. CONCLUSION

First, we made a compliance matrix for requirements verification with regards to the general CASE tools requirements defined in literature study and the proposed CASE tool in this study. The usability test of the developed CASE tool application is an ongoing research at the case company. The research team are in the process of interviewing different stakeholders and IT developers about the results of utilizing the CASE tool as a replacement of the traditional development and maintenance processes. The observations already proves that the CASE tool will perform positively in the following areas: clear configurator knowledge, fast and efficient maintenance, less required time and expert resources both for development and maintenance, and fast and comprehensive documentation and maintenance.

REFERENCES

- [1] L. Hvam, N. H. Mortensen, and J. Riis, *Product customization*. Berlin Heidelberg: Springer, 2008.
- [2] L. Hvam, S. Pape, and M. K. Nielsen, "Improving the quotation process with product configuration," *Computers in Industry*, vol. 57, no. 7, pp. 607–621, Sep. 2006.
- [3] T. D. Petersen, "Product Configuration in ETO Companies," in *Mass customization information systems in business*, 2007, pp. 59–76.
- [4] L. Ardissono *et al.*, "A Framework for the Development of Personalized, Distributed Web-Based Configuration

- systems," *AI Magazine*, vol. 24, no. 3, p. 93, 2003.
- [5] F. Salvador and C. Forza, "Configuring products to address the customization-responsiveness squeeze: A survey of management issues and opportunities," *International Journal of Production Economics*, vol. 91, no. 3, pp. 273–291, Oct. 2004.
- [6] S. Shafiee, "Conceptual Modelling for Product Configuration Systems," Technical University of Denmark, 2017.
- [7] S. Shafiee, K. Kristjansdottir, L. Hvam, and C. Forza, "How to scope configuration projects and manage the knowledge they require," *Journal of Knowledge Management*, vol. 22, no. 5, pp. 982–1014, 2018.
- [8] C. Coronel, S. Morris, and P. Rob, *Database systems: design, implementation, and management*. Cengage Learning, 2010.
- [9] D. L. Kuhn, "Selecting and effectively using a computer aided software engineering tool," Westinghouse Savannah River Co., Aiken, SC (USA), 1989.
- [10] S. Shafiee, L. Hvam, A. Haug, M. Dam, and K. Kristjansdottir, "The documentation of product configuration systems: A framework and an IT solution," *Advanced Engineering Informatics*, vol. 32, pp. 163–175, 2017.
- [11] L. Hotz, A. Felfernig, M. Stumptner, A. Ryabokon, C. Bagley, and K. Wolter, *Configuration Knowledge Representation and Reasoning*. Elsevier Inc., 2014.
- [12] A. Duffy and M. Andreasen, "Enhancing the evolution of design science," in *Proceedings of ICED 95*, 1995.
- [13] P. Y. Chao and T. Te Chen, "Analysis of assembly through product configuration," *Computers in Industry*, vol. 44, no. 2, pp. 189–203, 2001.
- [14] D. Magro and P. Torasso, "Decomposition strategies for configuration problems," *Ai Edam*, vol. 17, no. 1, pp. 51–73, Aug. 2003.
- [15] A. Haug and L. Hvam, "Representation of Industrial Knowledge-as a Basis for Developing and Maintaining Product Configurators," Technical University of Denmark, 2008.
- [16] M. Aldanondo, S. Rouge, and M. Ve, "Expert configurator for concurrent engineering: Caméleon on software and model," *Journal of Intelligent Manufacturing*, vol. 11, no. 2, pp. 127–134, 2000.
- [17] Tseng, H.E., C. C. Chang, and S. H. Chang, "Applying case-based reasoning for product configuration in mass customization environments," *Expert Systems with Applications*, vol. 29, no. 4, pp. 913–925, Nov. 2005.
- [18] J. Tiihonen, M. Heiskala, A. Anderson, and T. Soininen, "WeCoTin – A practical logic-based sales configurator," *AI Communications*, vol. 26, no. 1, pp. 99–131, 2013.
- [19] K. Kristjansdottir, S. Shafiee, H. Hvam, C. Forza, and N. H. Mortensen, "The main challenges for manufacturing companies in implementing and utilizing configurators," *Computers in Industry*, vol. 100, no. May, pp. 196–211, 2018.
- [20] F. MacDonald, J. Miller, A. Brooks, M. Roper, and M. Wood, "Automating the software inspection process," in *Computer Aided Software Engineering*, 1996, pp. 9–34.
- [21] L. Fowler, M. Allen, J. Armarego, and J. Mackenzie, "Learning styles and CASE tools in Software Engineering," in *9th Annual Teaching Learning Forum*, 2009, pp. 1–10.
- [22] V. E. Barker, D. E. O'Connor, J. Bachant, and E. Soloway, "Expert systems for configuration at Digital: XCON and beyond," *Communications of the ACM*, vol. 32, no. 3, pp. 298–318, Mar. 1989.