Citation for published version:
Schaefer, D 2004, 'A New Method for Variant Design Technology in ECAD' Paper presented at ECAD/ECAE2004 – 1st International Conference on Electrical/Electro¬mechanical Computer-Aided Design & Engineering, Durham, UK United Kingdom, 15/11/04 - 16/11/04, pp. 46-49.

Publication date:
2004

Document Version
Publisher's PDF, also known as Version of record

Link to publication

University of Bath

A New Method for Variant Design Technology in ECAD

D. Schaefer
School of Engineering, University of Durham, UK

Abstract
One of the most important approaches to enable cost and time reduction with respect to Computer-Aided Design for electrical/electromechanical engineering (ECAD/ECAE) is to develop, generate, and handle design variants in an efficient manner. The objective of this paper is to present a generic variant design technology approach that has a high potential for the efficient reusability of existing projects. More precisely, the paper will present the procedure to allow the new methodology in variant design technology to be implemented within an arbitrary ECAD environment. The method presented automatically generates a complete technical documentation of an electrical installation on the basis of a placed order specification. This involves three major steps. Firstly, a product variant of an installation is configured on the basis of existing standardised modules. Secondly, based upon the corresponding configuration file, a set of commands describing the generation of a typical ECAD project containing the configured modules is automatically compiled. This is a key novelty, as all commands are expressed in a non-system specific meta-language, which can then be automatically translated into a macro programming language of a specific ECAD system and stored as a file. Thirdly, the specific ECAD system can import and process the file to create a practical ECAD project of realistic complexity.

Keywords:
Variant Design, ECAD/ECAE, Process Automation, Product Configuration

1 INTRODUCTION

In order to remain competitive in the market, most companies offer their products in a range of different variants. However, new variants of existing products are more and more often composed of existing basic components rather than newly designed. A precondition to allow for this is to have modular product structures. An industrial branch where this is common practice is plant engineering and construction. The majority of companies developing electrical/electromechanical installations tend to reuse their existing designs, plans and project documentations to develop additional customer specific variants.

Unfortunately, this process is predominantly performed manually, which is far from being efficient and effective. Due to the permanently increasing pressure of competition it is vital to develop automatisms that facilitate the generation of product variants.

The basic idea behind the variant design approach presented in this paper is simply to automate the workflow process of creating ECAD variants as it is manually performed by most companies today. The approach is closely related to industrial best practice and derived from day-to-day operations.

1.1 The basic concept as applied in practice

Many companies approach their potential customers by technical field sales and distribution staff. These sales persons usually have selling catalogues on their disposal, which allow customers to assemble bespoke product variants based on basic modules or components that can be combined. Hereby, the number of components that can be chosen from (at this stage of the sales process) tends to be relatively small.

Once a rough pre-configuration is finished, design engineers and other office employees get together to determine a resultant technical fine configuration. This fine configuration comprises all parts and components necessary to make up the product variant desired and – depending on the nature of the modular product structure associated – may consist of thousands of items. This fine configuration also includes technical knowledge in terms of rules and constraints pinpointing under what circumstances what components may or may not be combined with others.

In order to identify specific components or items, designers usually refer to part numbers, drawing numbers, BOM entries, or similar terms. Specifically in the area of electrical/electromechanical engineering most of these configuration items are schematic diagrams, terminal plans or a variety of other documents required to express functionality and assembly of an installation.

The next step in creating the configured variant requires a design engineer to start an ECAD system, generate a new project, copy the plans configured into the project and specify individual customer and order details. Subsequently, alterations necessary to the specific project have to be accomplished and the process of generating an updated complete electrical documentation according to the alterations or amendments made has to be initiated.

In order to make the development of product variants more effective, ECAD system vendors aim towards an automatic computer-aided support of the workflow process outlined above.

2 VARIATIONAL DESIGN TECHNOLOGY APPROACH

2.1 Functional Principle

The basic functional principle of the variant design approach presented in this paper is now described. It is based on aspects from knowledge-based product configuration, the programming of variants, as well as parametric modelling and process automation. The fundamental idea behind the approach is to automatically generate an entire technical documentation of an electrical/electromechanical installation on the basis of a placed order specification. The overall process to achieve this involves five steps as shown in figure 1.

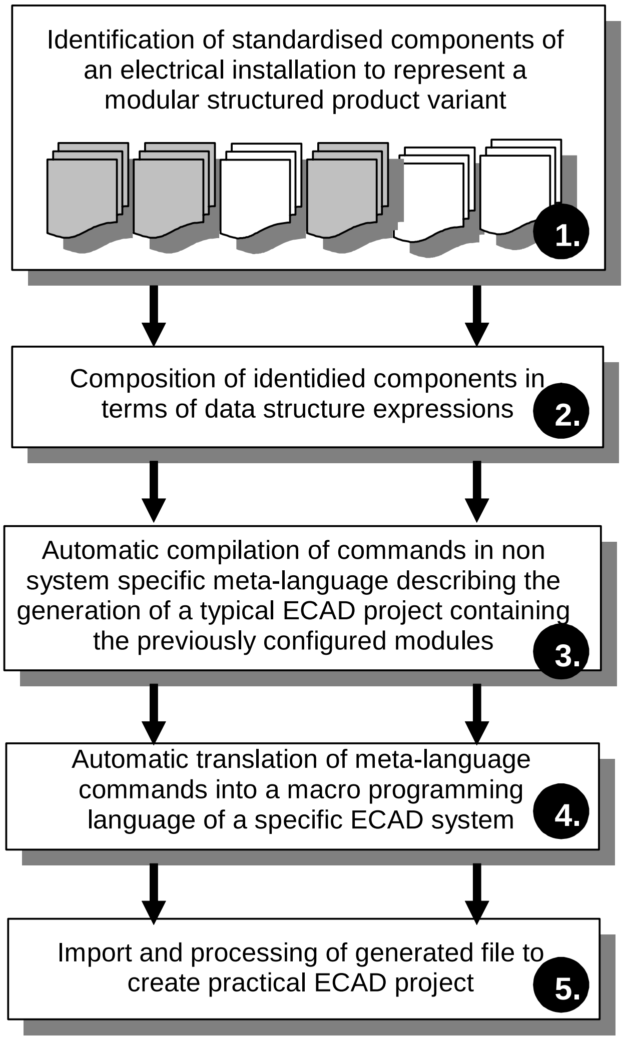

Figure 1: Functional principle of ECAD variant design technology approach.

Firstly, based upon a modular product structure a variant of an installation is configured by composing standardised modules of existing (previously developed) components.

Secondly, all components identified to compose a specific variant configuration are stored in a data file. This includes settings for parameter values that may finally define some characteristics of the components. This configuration file corresponds to a bespoke data structure newly developed and appropriate to describe ECAD product variants in general.

Thirdly, based upon the configuration file created, a set of commands describing the generation of a typical ECAD project containing all the components configured together is automatically compiled. This is a key novelty, as all commands to generate such a project are expressed in a non system specific meta-language.

Fourthly, the variant project description expressed in non system specific meta-language commands is automatically transformed into a batch file of commands of a macro programming language of associated to a particular ECAD system.

Fifthly, the specific ECAD system imports and processes this batch file to create an actual ECAD project. This project then may be handled or dealt with in any way the ECAD system allows.

2.2 Technology Approach

A technical approach to realise the functional principle is now given. An overview is illustrated in figure 2. Since step one of the functional principle is obvious, step two is the point to start with going into further detail.

As already mentioned in paragraph 2.1, a data structure specifically tailored for representing configuration based variant projects is required. This data structure primarily has to cover components and documents typically used to make up an entire project documentation of an electrical installation. Furthermore, the data structure it has to span semantic information on relations and constraints existing between components or documents. In order to be able to store, import, process and automatically evaluate variant projects based on a semantics-based data structure in subsequent processes, a standardised data format for representing the data structure has to be chosen. In the approach described in this paper the variant project data structure has been expressed in terms of a document type definition (DTD) associated to the language XML (extensible mark-up language). Simply speaking, XML is a mark-up language for describing hierarchically composed objects that strictly distinguishes between structure, content and layout of the objects data. The specific purpose of a DTD is to define the structure of XML files.

Based upon these technical arrangements, a variant project expressed in terms of XML can be validated against a DTD to check whether or not its structure complies with the requirements defined in the DTD. So far, configuration based variant projects (components, documents, relations, constraints, etc.) whose original underlying data structure has been transferred into a XML pendant can be stored as XML data files to be further processed later on.

Realising step three of the functional principle involves formulating commands expressing the generation of a configuration based ECAD project in a non system specific form. Hence, a suitable meta-language has been developed and applied. This meta-language covers a variety of system commands similar to those being typical for contemporary ECAD system's macro languages.

Following the functional principle of paragraph 2.1, a variant project stored as XML file now can be transferred into a new data format describing the generation of an ECAD project containing the configured components.

Technically speaking, this means to enhance the original XML data structure of a configuration based variant project in such a way that it allows to model and express commands in the meta-variant-construction language 'vcl'.

In order to allow for further standardised data processing operations to be performed on vcl files, the data structure of the meta-language vcl has to be transferred into an equivalent XML DTD. Henceforth, this meta-language expressed in terms of an XML DTD will be denoted as variant construction language in XML, in short 'vclX'. To sum-up, the components of a configured ECAD variant project stored as XML file have been picked-up and transferred into another, more sophisticated and powerful

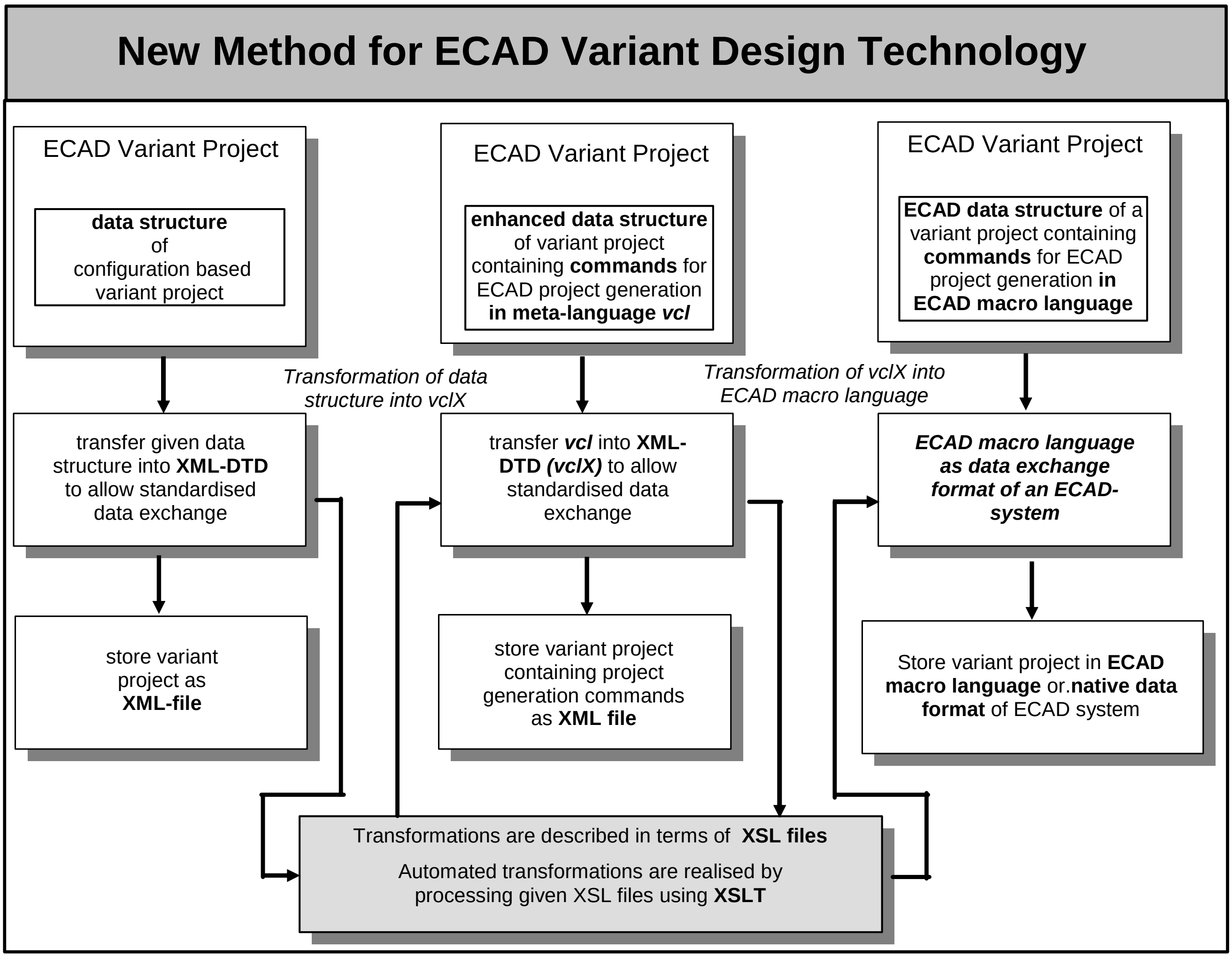

Figure 2: Approach to an automatic generation of ECAD project

XML data structure containing non system specific commands to describe the generation of an ECAD project made up of the components configured.

The above transformation process is carried out automatically using 'XSL' (Extensible Style sheet Language) and a software tool called 'XSLT' (XSL Transformation). XSL is a language specifically developed to facilitate transformation purposes and allows defining rules describing the transformation from one XML structure into another data structure. The transformation rules necessary to perform the desired transformation are stored within a specific XSL data file. The actual data transformation then is carried out by XSLT which requires three things to work: (1) a source XML file to be transferred, (2) the source file's corresponding DTD and (3) a XSL file containing all the rules describing the mapping onto the target XML data structure.

The procedure described above analogously recurs for step 4 of the technology approach. However, this time an XSL file describing the transformation from the non-system specific vclX command list structure into another structure encompassing commands of a specific ECAD system's macro language is required. The result of this final transformation is a batch file to be imported and processed by the specific ECAD system chosen. In other words, a real ECAD project in a native data format has been created.

2.3 Module Conception

In this paragraph a module conception for implementing the approach discussed above is presented. It is tailored to support a workflow closely related to that described in paragraph 1.1. For an overall illustration of the module conception see figure 3.

The variant module basically comprises of two sub-modules, notably the 'configuration module' and the 'coupling module'. It is developed as a self-contained unit that may be coupled to one or more ECAD systems rather than directly implemented within a specific system.

The purpose of the configuration module is to perform the tasks carried out by design engineers and office employees as described in paragraph 1.1. This means to either create new configurations or adjust existing ones based upon a modular product structure using existing basic parts or components. Hereby, the various plans of an electrical documentation (e.g. schematics, terminal plans, part lists, etc.) are drawn on as configuration objects.

Prior to being able to work with the configuration module all the components (projects, sub-projects, etc.) existing on ECAD site and meant to be available for variant projects have to be imported to the configuration module's database.

Using the features of a comfortable graphical user interface (GUI), the imported ECAD components can be used to combine new variant projects, to adjust previously created variants and to store further basic components into the database. The design knowledge regarding rules describing possible combinations and configurations has to be brought into the database as well. The system kernel of the configuration module therefore has to incorporate an intelligent mechanism to maintain, check and control the compliance of constraints modelled.

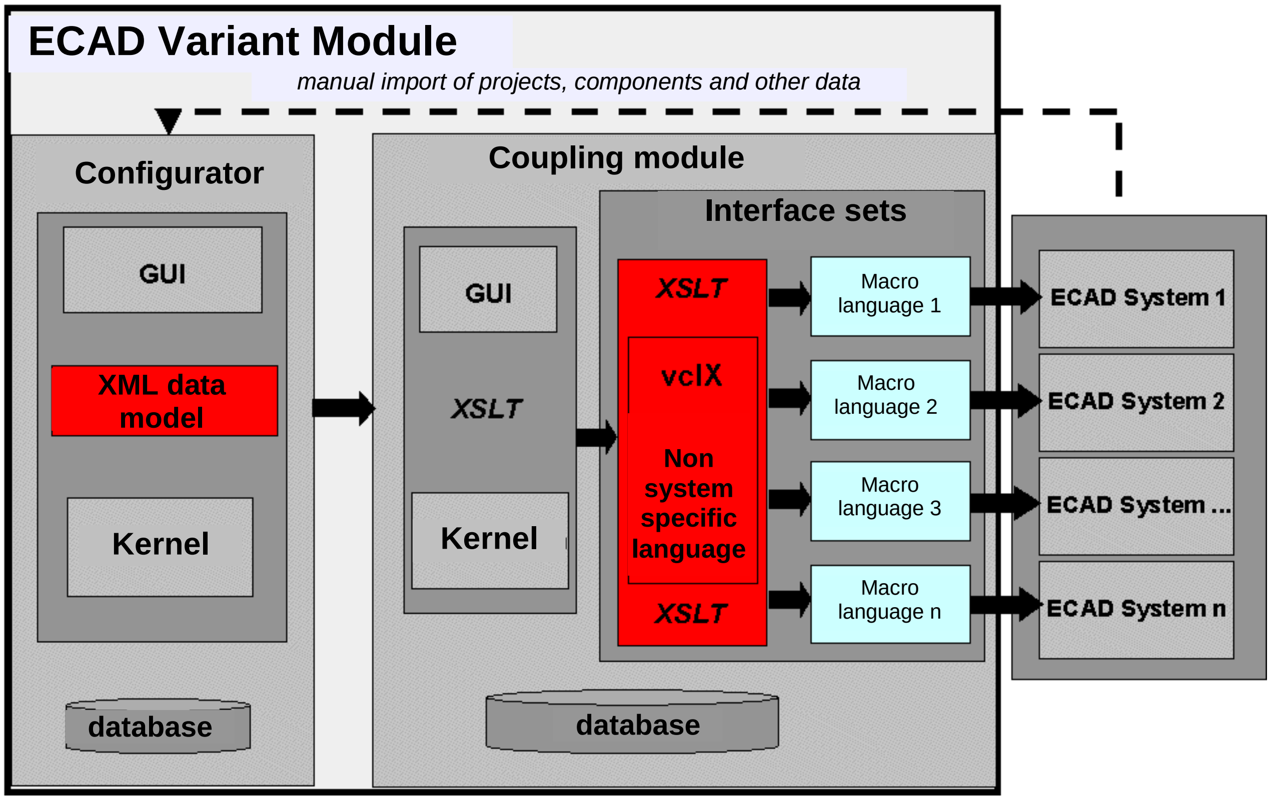

Figure 3: System architecture of the variant module

Due to the complexity of knowledge based configuration systems the development of proprietary knowledge based configuration tools is not recommended. There are many commercial solutions for almost any configuration tasks available on the market.

The purpose of the coupling module is to automate those steps of the workflow process that today are usually performed manually by an engineer after the components for a variant configuration have been determined.

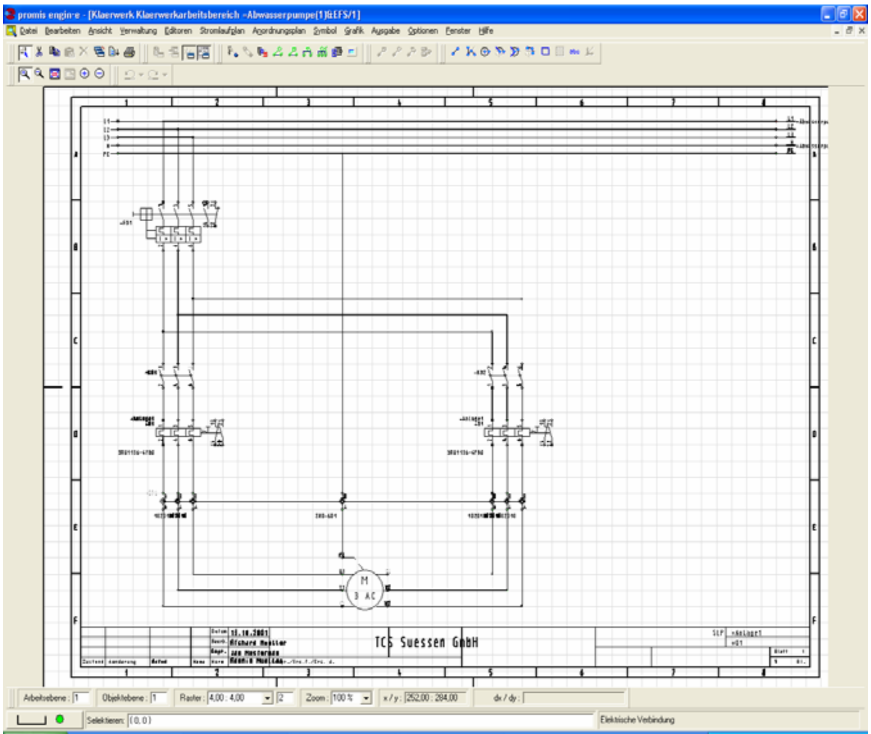

Figure 4: Example of a simple schematic automatically generated with the variant module

A first task for the coupling module is to import a variant configuration from the configuration module. Subsequently, it has to create a batch file of non-system specific commands describing the relevant steps to open a new ECAD project and to include the chosen components. After that, the coupling module has to transfer this non-specific commands into a batch file for a specific ECAD system which can import and process the final batch file then.

3 CONCLUSION

The approach presented in this paper has been realised as a software prototype. Its general applicability has been proven and several possibilities to amend, extend, or standardize the suggested technological approach are currently being investigated. Commercially enhanced, a variant module like the one proposed in this paper would bear a high potential in respect to cost and time reduction.

4 REFERENCES

[1] Schaefer, D., 2003, Variantentechnologie unter besonderer Berücksichtigung von Elektrotechnik CAD, Shaker-Verlag, Aachen, ISBN 3-8322-1901-3, 292 pages