The paper presents concepts for the integration of software components as CAD systems, document management systems and expert systems into a tool box for the electrical engineer, and describes the architecture of the resulting system for computer integrated engineering (CIE). The kernel of the tool box is an engineering data base system based on an active DBMS for maintaining the logical integrity of engineering data. The schema of the engineering data base consists of a pre-defined kernel schema of commonly used objects for electrical engineering and a user definable part. In order to enable user modelling, a CIE modelling system is available for defining application specific data structures, relationships and engineering rules.

1. Introduction

Today's engineering projects get ever bigger and more complex. At the same time, the flexibility of the engineer with respect to price, duration, quality standards and deadlines is narrowed. Therefore, the engineer needs to be supported by suited tools and an integrated environment of such tools in a way - that the resulting tool box can be adapted to the specific needs of the user, - that each step of the engineering work is supported by the system, - and that all outcoming data can be used by subsequent tasks.

The CIE (computer integrated engineering) system for electrical engineering of ABB CADE International is a tool box fulfilling these requirements.

Chapter 2 describes the required functions of such a system, chapter 3 gives an overview of the architecture, chapter 4 shows the involved components, and chapter 5 gives an outlook of planned developments.

2. Requirements

Engineering consists of 2 phases:
- In order to offer a system or plant, the structure of the technical solution must be found, the main functions have to be identified, the costs of products and services are calculated and the needed resources planned.
- After the project has been ordered by the client, the engineer must be supported in project planning, project control, configuration of the plant or product, electrical and mechanical layout, and resource management. Consequently, a great number of documents results, e.g., the confirmation of the order, user documentation, manufacturing documents, need of material, and others.

The engineer must be supported in the diverse functions by a uniform user interface. The tool box must support the following functions:
- management of client data, as well as project data
- document management and document composition (text and graphics),
- management of component and other technical data as variants, standards, and technical characteristics,
- product configuration and cost calculation according to the technical specifications of the client,
- drawing of the wiring and cable plans,
- management of macros and standards for the drawings,
- technical calculations and simulations,
- interface to the MRP system.

3. Architecture of the ABB CADE CIE system for electrical engineering

After analyzing the described functions, the conclusion must be drawn that the different clients of such an integrated system demand a variety of different shapings of the CIE system modules. As a result, we did not build a functionally integrated system for all mentioned requirements, but an extendible and adaptable tool box with uniform data management and uniform user interface.

Kernel of the tool box is the engineering data base which is used by all tools for the management of common data or data to be exchanged. Staley and Anderson [StAn 86] list a number of requirements to an engineering data base management system. The most important ones are
(1) support of multiple engineering applications
(2) dynamic schema modification and extension
(3) support of design iteration
Requirement (2) has been the most important one for the design decisions concerning the architecture of our CIE tool box (fig. 1). There does not exist a common, widely accepted schema covering all different applications of electrical engineering. Therefore, we decided to support a common, mandatory kernel schema of base objects and the extension of this object by the client who can model in this way his own application schema. For this purpose he is supplied with a so-called CIE modelling system, i.e. a data dictionary system (requirement 4) for defining data structures, relationships and engineering rules (requirements 5 and 6).

In order to maintain the logical consistency of the data, i.e. to enforce the user-defined constraints, an engineering data base management system is needed, that allows to define and to guarantee the obedience of arbitrary consistency constraints. For this purpose, we chose the active database shell KIDS [WuBB 89] that implements an event trigger mechanism [DiKM 86] for consistency control on top of an arbitrary relational DBMS.

Not all data of the different tools in the CIE tool box are kept in a central data base. Otherwise, the engineering data base -- especially one built on top of a relational system -- would be the bottle neck of the whole CIE system. Data that is local to one single tool (e.g. the graphical layout of CAD drawings) is kept internally by that specific tool. The existence of such data however is recorded in the engineering data base (for instance in order to consider it in archive operations). Furthermore, there will be a number of tools, that for the same reasons do not read (resp. write) common data (e.g. the logic contained in CAD drawings) directly from (resp. into) the engineering data base, but use instead a check-in/check-out mechanism [LMPD 85]. If such tools have a flexible data schema (e.g. the expert system shell KEN - see below), the check-in/check-out DB-interface must be generated out of the CIE modelling system. If they rely on a fixed data schema (e.g. a document composition system), the check-in/check-out DB-interface is fixed too.

4. Components of the CIE tool box
The most important tools of the system environment are the following (fig. 2):
The CIE Modelling System

The modelling system consists of an editor to formulate data structures, relationships, actions and triggers according to the CIE meta schema. Objects belonging to the kernel schema may not be changed. Interfaces are supported to the engineering data base that is used for storing and retrieving the schema, to all tools that are build around a flexible data schema, and to the check-in/check-out interface. There exist also interfaces from the tools schemas to the CIE modelling system. These interfaces come into action when a client has already used one or another tool before introducing the whole environment. This allows a stepwise extension of the engineering tool box, starting even without the common DB (e.g. with a stand-alone CAD or expert system only).

The Active Data Base System KIDS

KIDS supports -- beside the functionalities provided by the relational DB kernel (e.g. ORACLE) as physical data integrity, backup/recovery, concurrency and data protection -- an event trigger mechanism to maintain the logical data integrity. It therefore controls the technical rules and conditions that are often present in engineering tasks.

The Engineering Data Base

The engineering database consists of two parts, namely project data and standards data. Each data element is either part of a project or part of a catalogue. Data exchange between projects is implemented as making a physical copy of the data. References between information of different projects are not supported. In contrast, catalogue data used in a project is stored as a logical reference to the catalogue (no redundancy, fully up-to-date). Later, an extended version of the tool box will provide version control for catalogue data as well as for project data. The data base schema is outlined in fig. 3.

Expert Systems for Configuration and Calculation (KEN based)

In order to build such expert systems, the frame based expert system shell KEN is used [Viti 86]. A configuration system built on top of KEN supports the interactive definition of technical specifications of the client and derives (among others) a specification of the needed products, components or variant parts and services to fulfil the technical specifications, part lists, project data, and cost and calculation data.

The CAE-System DDS-C for electrical engineering

DDS-C is a wide spread, efficient and flexible CAD-CAE-System for electrical engineering. DDS-C supports the drawing of wiring and cable schemes for plants, machines and similar electro-mechanical systems, and their evaluation. Due to its open architecture, the system is well suited for the integration in the CIE environment.

The Document Management System DMS

DMS supports a structured and secure description, storage and retrieval of documents, drawings and other technical information media. It consists of the following subsystems:

(1) management of classification systems,
(2) navigation in classification (poly-)hierarchies,
(3) definition component for technical indices (attributes) per class, incl. inheritance of these attributes to all subclasses of the classification (poly-)hierarchy,
(4) management of documents (attachment to the classification, definition of attribute values),
(5) retrieval by attribute values,
(6) data protection,
(7) archive functions, and
(8) release mechanisms.

Interface programs

Interfaces to the most important MRP-, NC and test systems available on the market.
Check-In/Check-Out - Interface to the Engineering Data Base

During check-in all engineering rules must be checked that can not be guaranteed by the tool making check-in to the database itself. Inconsistent data is either not admitted to the database or marked as inconsistent, thus disabling its use in further engineering phases. During check-out of data by an engineering tool making use of this data over a long period of time (longer than minutes), the information read in the database is marked by a so-called intention lock, noticing all other prospective users that the data in question is in use.

5. Conclusions

The project is currently in the phase of its detail specification. Most of the described components are already available as stand-alone applications.

After finishing the integrated tool box, the available data base systems shall be re-evaluated. Especially the question must be answered whether there will be new data base systems for non-standard applications (e.g. object oriented DBMS [AnHa 87]) ready for a wide spread use in practice. We await much more efficiency in manipulating complex data structures by such a system. This would allow to interface all CIE tools directly to the database instead of using the check-in/check-out mechanisms.

The user of the CIE tool box gains important advantages in terms of project time, quality, costs, and transparency (despite the ever increasing complexity of engineering projects) by the integration of all needed functions in a uniform tool box, by the redundancy free data handling in an engineering data base and its use throughout all engineering phases, and by making use of modern techniques such as expert systems and active data bases.

References


