Customization of the CAD Software in a Typical Drawing Office for a Power and Electricity Distribution Company in Zimbabwe

W. R. Nyemba[1], C. Mbohwa[2]

[1]Department of Mechanical Engineering Science, Faculty of Engineering & the Built Environment, University of Johannesburg, Auckland Park 2006, Johannesburg, South Africa

[2]Professor of Sustainability Engineering, Department of Quality and Operations Management & Vice Dean for Research and Innovation, Faculty of Engineering and the Built Environment, University of Johannesburg, Auckland Park 2006, Johannesburg, South Africa

(nyemba@yahoo.com)

Abstract – **Computer Aided Design (CAD) packages are normally bought off-the-shelf for a wide range of engineering disciplines. However, a single system cannot possibly meet requirements of different users as these packages are normally provided with basic primitives but no blocks for quick generations. A case study carried out at a power enterprise in Zimbabwe revealed delays in attending to faults and installations at their substations partly because of delays in the provision of working drawings. Although the company migrated from manual to CAD drawings, the techniques they employed resulted in duplications and delays. An in-house software development strategy was employed in this research to customize their AutoCAD software through an industrial engineering approach aimed at lowering man-hours by generating and storing symbols of their equipment for retrieval and use in future drawings, resulting in lead time reductions and integration with their maintenance management system to avoid duplication of tasks and information.**

Keywords - **AutoCAD, attributes, blocks, CAID, customizing**

I. INTRODUCTION

CAD packages can be purchased off-the-shelf or developed in-house by the end user [1]. The latter is usually capital intensive and can be quite prohibitive but results in very specific software that caters for the users' needs. Owing to the expenses involved in in-house developments, more companies are turning to Commercial Off-The-Shelf (COTS) software or turnkey systems by purchasing complete systems from vendors [2]. Dedicated and experienced specialists develop turnkey systems such that these have a high reliability and many such systems come with a warranties [2]. To ensure commercial viability for the developer, these systems are generic and designed to suit as large a market as possible. However, a standard turnkey system is unlikely to meet every single requirement of the end user. Apart from costs and time to develop in-house software, benefits are usually long-term and in some cases unjustifiable for small businesses. Generally, the idea would be to purchase a COTS software which satisfies the maximum number of requirements for the company while the rest can accomplished by customizing it by way of add-on facilities [3]. Whereas the trend by most companies is to go for COTS software, the need to customize a software package arises from the deficiency of turnkey systems in that they are generic and the rate of software updates can be equally costly. Custom developed software is easier and allows for the growth of the system in tandem with the business resulting in efficiency, productivity and quality of service [4].

The power and electricity distribution company produces a wide range of engineering drawings for their substations and other engineering projects in Zimbabwe. Traditionally they produced drawings manually but computerized their Protection Drawing Office (PDO) for consistent and accurate drawings as well as speeding up the production, driven by the demand from their various sites. The company was thus faced with a number of challenges on transition, firstly the need to get their engineers trained in CAD and secondly to tailor make their CAD software to enable them to produce the required drawings with ease while utilizing the essential elements of the software. Most users of AutoCAD are familiar with its drafting capabilities but little else and yet the various options and techniques such as the use of blocks and attributes is less explored when in fact these are what really makes the difference [5]. The company employs over 5,000 workers in its 4 engineering subsidiaries, making it a large enterprise anchored on provision of electricity and technical support throughout the country [6]. The company has been growing, with several substations constructed throughout the country as well as new ventures to increase energy on the national grid owing to the increased demand and hence the increased need for technical support and thus the requirement for the rapid production of engineering drawings by its subsidiaries.

It is within this context that this research set out to balance between the use of COTS and in-house developed software needed to meet these demands, resulting in a new approach for Customized Add-on In-House Development (CAID) based on both aspects while utilizing an academic approach to contain costs. This process and the transition from manual drafting needed to be managed carefully and hence the industrial engineering approach in this paper that treated the development of the CAID software as a process separate from particular applications to any of the company's specific projects [7]. This paper outlines the process through which the CAID software was developed and superimposed on their AutoCAD software for the company's specific applications in the rapid production of drawings required at their various substations and projects.

II. CUSTOMISATION METHODS AND STRATEGIES.

Rapid changes in technology invariably prompt rapid updates in software and in some cases development of new ones [8]. This is evidenced by the number of changes made by Autodesk since they released the first version of AutoCAD in 1982, now with almost a new version every year but back then it was after every 2-3 years [9]. In general however, there are several approaches that software developers have adopted over the years in order to customize or develop new software. A commonly used and rapid method for software development is the agile software development technique which involves development of the software through iterations, typically lasting one to four weeks and each iteration exemplifying a small software project in its own right [10]. Agile Software Development (ASD) can be carried out through methodologies such as crystal methods, Dynamic Systems Development Model (DSDM), Scrum and Extreme Programming (XP) [11]. This methodology was developed mainly to resolve problems encountered in traditional software development such as the limitations of and difficulty in changing the users' existing software to incorporate the additional changes [12].

Although ASD is an emerging and popular approach that enables practitioners to rapidly develop software, development of software using this approach requires active user involvement and close collaboration during the development cycle. Whereas these are critical success factors, it however means users invariably have to give up their time to commit themselves full time in the development process and this will obviously have an impact on the organisation's business. The very nature of ASD is in the agility during the development process and the flexibility to change direction. This however poses potential challenges such as the risks of a 'never-ending' project leading to much less predictability [12]. The use of the Scrum or Kanban approaches in software development optimizes the processes by identifying the tasks, managing time more effectively and setting up teams. However these approaches may be more convenient for small to medium enterprises comprising of small teams as they are good for small and fast moving projects [13].

AutoCAD was developed with an "open architecture" that allows for modification of the standard package [14]. Many of the characteristics of AutoCAD's user interface can be modified without leaving the CAD environment. It is also possible to create project directories where AutoCAD will search for various files before declaring them as missing. Within the AutoCAD environment, tools can be copied from existing toolbars into a user defined toolbar such that those tools that are most frequently used are available on a single toolbar. AutoCAD also allows users to design new toolbars completely from scratch. Users can also add their own menus to the AutoCAD pull down menus which can be used to load List Processing (LISP) routines and so forth [15]. In addition, menus can be used to launch favorite or frequently used commands and to execute macros.

AutoCAD provides a script facility that allows commands to be read from a text file. This feature allows a user to execute a predefined sequence of commands to automate repetitive tasks. AutoLISP is a high level interface language for AutoCAD derived from the LISP programming languages. AutoLISP allows users of AutoCAD to write macro-programs and functions that are well suited for graphics applications using a built-in LISP interpreter that is used to enter the AutoLISP code at the command prompt or to load it from external files [15]. AutoLISP applications or routines can interact with AutoCAD in many ways to prompt the user for input, access built-in AutoCAD commands directly, and modify or create objects in the drawing database. ActiveX Automation in AutoCAD provides a mechanism to manipulate AutoCAD programmatically from within or outside AutoCAD by exposing various AutoCAD objects to the "outside world". Once these objects are exposed many different types of programming languages and environments such as Microsoft Word or Excel can access them. Drawings can also be exported in AutoCAD's Drawing Web Format (DWF), a highly compressed 2D-vector file that can be used to publish an AutoCAD drawing on the World Wide Web using the commonly used web browser or through plug-ins provided by Autodesk to enable others to view the drawings via the Internet. This is an area of application that has vast potential for industrial engineering and marketing of engineering products as the Internet continues to play an increasingly important role in advertising [16].

Whereas the AutoCAD environment allows users to input commands and function prompts through the command line, dialog boxes simplify this process by providing user specific prompts and are therefore desirable as they have a very strong visual appeal. The Dialog Control Language (DCL) in AutoCAD provides a convenient interface for getting user input to add-on functions while AutoLISP controls the functionality of the dialog boxes. The main focus for this paper is on the use of blocks and attributes in customizing the add-on facility to enable the company to produce drawings more rapidly. Blocks have the advantage of reducing sizes of drawings because multiple instances of blocks are stored in one reference in the drawing database and it is also possible to create a symbol library containing commonly used blocks as a way of ensuring standardization across a series of drawings. An attribute is a drawing entity designed to hold text and to link text data such as part number, supplier or comments pertaining to the block to graphic objects in the drawing database [14]. Attribute information can be extracted from a drawing to compile a spreadsheet or as an input to a database to produce bills of materials by making use of the American Standard Code for Information Interchange (ASCII) file in the form of templates that contain the attribute tags to be extracted [17]. This paper focusses on addressing the company's failure to attend to breakdowns at their various substations and projects throughout, owing to delays in the provision of the requisite drawings. Having established that the case study company is a large enterprise and that as much as possible the development of the CAID must not disrupt operations at the company, the use of ASD and associated approaches to develop the required software was shelved, preferring an add-on facility that will not only ensure that the company makes use of available software such as AutoCAD but there will be minimal disruption of operations during the development process.

III. CASE STUDY AND METHODOLOGY

The core business of the power and electricity distribution company is to provide electricity to consumers throughout the country, mechanical and electrical engineering thus being the major functions of the company's operations. Drafting is a major component of the work done in the PDO. The department is responsible for designing the security features that guard against system overloads and the drawings produced are mostly schematic diagrams that show design and installation details for the wiring at substations. The PDO contains a multitude of electrical drawings that consist of many symbols representing the actual components and pieces of equipment, hence the need to clearly distinguish them in line-drawing images [18]. The company has a countrywide computer network which is used to control the national grid and to report any failures to the national control center. When a failure occurs on the grid, this is recorded instantly at the control center and the maintenance required is effected through their repair channels and the established Maintenance Management System (MMS) Physical Assets Module.

The case study of the drafting operations at the company revolved around four major issues that warranted the development of a system to deal with the challenges. Firstly there was need to automate the creation of symbols in order to produce drawings rapidly. The traditional methods used by the company to create and insert the symbols in drawings were quite inefficient, sometimes involving duplications. Because AutoCAD allows a single drawing to be open per session, this procedure meant running multiple sessions of AutoCAD simultaneously. Apart from delays in producing the eventual required drawing, this also increased the demands for computer memory. The non-graphic data associated with the symbols could not be automatically retrieved from the drawings. Thus, if this information was required for compiling bills of materials, this was done manually, hence the need for creating the bills of materials automatically. When new equipment is installed and commissioned, the PDO would be the first place where the installation is documented in the form of "As-Built" drawings. It made sense then, that this should also be the place where the first entry is made to the MMS Physical Assets Module in order to start compiling maintenance records for assets. Most computer users find it inconvenient to refer to an operation manual when using a software package, hence the need to provide an on-line help facility to make such information accessible without leaving the CAD environment [19].

Based on this scope, the requirements planning was derived from working directly with the engineers at the PDO, constantly engaging them and drawing from their experiences and needs. Fig. 1 shows the summary cycle through which the development of the CAID software at the company was followed. Continuous test runs were conducted as the customization progressed and improvements effected (dashed line) and also done in an open architecture to allow for continuous improvements after testing for functionality and usage.

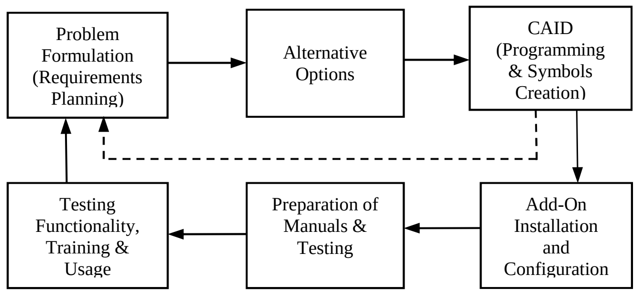

Fig. 1 CAID Methodology Cycle

Table I shows the alternatives that were considered for solving the challenges outlined. The shaded cells show the options that were selected for developing the solution based on ease of use and consultations with the engineers and technicians at the company.

TABLE I
MORPHOLOGY CHART (ALTERNATIVE SOLUTION SCHEMES)

Function	Means of Achieving the Function		
Launch Routine	Command name at command prompt	Use of Toolbar	Use a Pull Down Menu
Get User Input	Command line input	Dialog box interface	
Symbol Creation	Draw symbols using a script	Use of 'local' blocks	Use 'global' blocks with attributes

Several programs that drive the add-on utility/pull down menu based on the selected means of achieving the 3 main functions as shown in Table I were written. The 2 main programs were written in AutoLISP, the first one to drive most of the routines in the add-on facility while the other one to drive the dialog boxes that accept the various attribute values. An online help facility was also added in in conjunction with each dialog box and the general usage. Because of the length of the various AutoLISP and DCL programs that were coded for this research, they have not been included in this paper, suffice to say, the results section that follows is derived from those codes. In addition to developing the dialog boxes and custom routines, the various symbols were also created as simple drawings and stored in appropriate databases depending on the area of application, Civil, Electrical or Mechanical bearing in mind that some symbols are used across the databases.

IV. RESULTS

The solution developed from the morphology chart led to the development of the company's Add-on Utility, which consisted of routines for the automatic insertion of symbols and for attaching attributes needed to identify the component represented or to hold the maintenance data associated with that piece of equipment. A second function allowed maintenance information to be attached while the Add-On Utility customizes the extraction of all the maintenance data in a drawing for later use in the MMS Physical Assets Module. The custom routines are accessed through a pull-down menu, ZESA as shown in the AutoCAD screen shot in Fig. 2.

Fig. 2. ZESA add-on utility (pull down menu)

The insertion of symbols from the database into a drawing is done by selecting the appropriate category and required symbol using an image tile menu as shown in Fig. 3. Another dialog box will appear to prompt the user to enter the attribute values for each component such as shown in Fig. 4 for the insertion of a transformer.

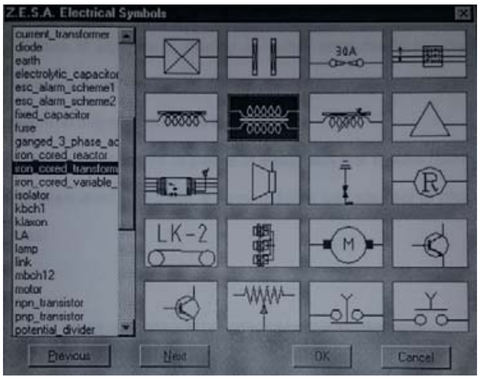

Fig. 3. Image tile menu for electrical symbols

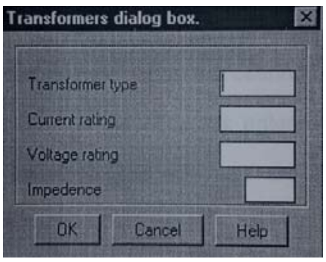

Fig. 4. Transformer dialog box attributes

If the component selected for insertion requires maintenance data, the dialog box shown in Fig. 5 enables the user to enter details for the Physical Asset Register.

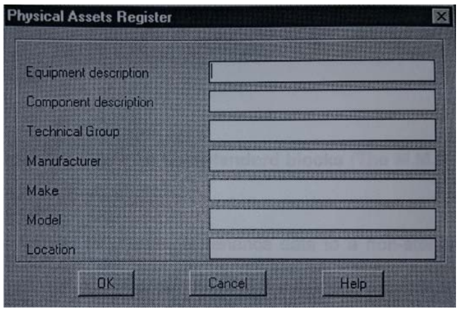

Fig. 5. Physical asset register dialog box

The exact location and size of the symbol on the drawing is controlled by the Insertion Point (*x, y, z* coordinates), Rotation and Scale on insertion, which wraps up the symbol insertion, complete with the definition of all the nongraphic data. When the attributes have been supplied, the Block name dialog box appears as shown in Fig. 6 prompts the user to supply a name for the block through selecting all the entities that make up the symbol and its insertion base point. The routine automatically defines selected entities as blocks, attaches the maintenance attributes to the block, and re-inserts the block in a required position.

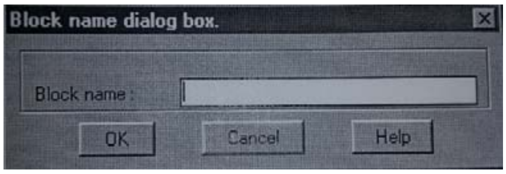

Fig. 6. Block name dialog box.

The Extract option under the MMS submenu launches the customized attribute extraction command. A dialog box appears that prompts the user to supply a name of the output text file to hold the extracted data. The attributes are exported to a space delimited text file that can then be used as input to the MMS Physical Assets Module, which is based on the Microsoft Access database management system.

V. DISCUSSION AND RECOMMENDATIONS

The results obtained from the customization work done at the company not only achieved the integration of AutoCAD with the company's MMS Physical Assets Module but also standardized their drawings for use throughout their entire network of substations and projects. Customization of the CAD system resulted in the reduction of reworks where maintenance information was entered manually, which is prone to errors. Technicians at the company no longer needed to refer to handbooks or manuals as the symbols could now be retrieved from the database together with their attributes. The definitions of attributes in the drawings were also standardized in that only the values of attributes are required to retrieve a symbol. The custom routines developed for the company enhanced the value of drawing files by including nongraphic information that can be used to automatically compile bills of materials that are required when teams attend to maintenance or breakdowns at the various substations. The estimated cost for the add-on facility was slightly over USD500 which was made up mainly of the programming time, a reasonable cost considering that software is generally expensive and thus this research saved the company a great deal.

In recent years, software upgrades have been effected at a faster pace than before. As such it was recommended that the company's CAD technicians need to be continuously developed and appraised of the changes in the new software so as to keep abreast and in pace with these changes. The work done in this research covered the company's PDO and most of the symbols used in this unit were captured. However the unit works in conjunction with other units of the holding company and thus it is necessary to standardize their systems. This research assisted the company and contributed in the rapid production of drawings and maintenance schedules required at the substations and projects countrywide, evidenced by the reduction in lead time between the reporting of faults and attending to them as well as the clearance of backlogs. However the major challenge and limitation to the work was sustainability of such a system in view of the rapid changes in the base software, AutoCAD, which would require the CAID software to be improved and updated with these changes regularly. This challenge was also in view of the fact that the customization was done through an academic approach which contained major costs that would ordinarily be levied, had it been done commercially. Further research into the work would be required to establish a sustainable system that can cope with changes in the base software, coupled with the continuous training of users.

VI. CONCLUSIONS

The custom routines developed for the power and electricity distribution company significantly reduced the number of keystrokes required for the creation and insertion of symbols in drawings, hence improving on productivity. Customized menus and symbols were created for use in the Protection Drawing Office resulting in the rapid production of drawings for use in maintenance at their various substations throughout the country. The use of blocks and attributes and other nongraphic information in conjunction with the CAD software allowed for the automatic production of bills of materials which would otherwise be manually accomplished from completed drawings. This also allowed for the integration of the company's CAD environment and their MMS Physical Assets Module to avoid duplication of information. The handling of nongraphic data was improved such that all the nongraphic information associated with symbols can now be retrieved automatically using standard AutoCAD commands or the custom Extract option.

REFERENCES

[1] S. Vinodh, S. R. Devadasan and C. Shankar, (2010), "Design agility through computer aided design", *Journal of Engineering, Design and Technology*, Vol. 8 Iss 1 pp. 94 – 106, 2010.

[2] T. Ahola, E. Laitinen, J. Kujala and K. Wikstrom, "Purchasing strategies and value creation in industrial turnkey projects", *International Journal of Project Management*, 26(2008), 87–94, 2008.

[3] W. T. Hsu, Y. Lu and T. Ng, "Does competition lead to customization?", *Journal of Economic Behavior & Organization*, 106(2014), 10–28, 2014.

[4] N. Basoglu, T. Daim and E. Sofuoglu, "A decision methodology for customizing software products" *International Journal of Industrial and Systems Engineering* 4(5):554-576, 2009.

[5] V. Quintana, L. Rivest, R. Pellerin and F. Kheddouci, "Re-engineering the engineering change management process for a drawing-less environment", *Computers in Industry*, 63(2012), 79–90, 2012.

[6] United States Energy Association, "Zimbabwe Power Sector", Available: https://www.usea.org/, Accessed: 20 July 2016.

[7] D. N. Card and R. A. Berg, "An industrial engineering approach to software development", *Journal of Systems and Software*, Vol. 10, Issue 3, 159-168, 1989.

[8] A. Kushwaha, S. K. Verma and C. Sharma, "Analysis of the concerns associated with the rapid release cycle", *International Journal of Computer Applications*, Vol. 52 No. 12, 20-25, 2012.

[9] Funding Universe, "Autodesk Inc. History", Available: http://www.fundinguniverse.com/company-histories/autodesk-inc-history/, Accessed: 31 July 2016.

[10] T. Dingsøyr, S. Nerur, V. G. Balijepally and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development", *Journal of Systems and Software*, 85 (2012), 1213– 1221, 2012.

[11] S. C. Misra and V. Singh, "Conceptualizing open agile software development life cycle (OASDLC) model", *International Journal of Quality & Reliability Management*, Vol. 32 Issue 3, 214 – 235, 2015.

[12] S. Misra, V. Kumar, U. Kumar, K. Fantazy and M. Akhter, "Agile software development practices: evolution, principles, and criticisms", *International Journal of Quality & Reliability Management*, Vol. 29 Issue 9, 972-980, 2012.

[13] H. Lei, F. Ganjeizadeh, P. K. Jayachandran and P. Ozcan, "A statistical analysis of the effects of Scrum and Kanban on software development projects", *Robotics and Computer-Integrated Manufacturing*, In Press, 2015.

[14] Autodesk, "AutoCAD 2013 user's guide", Autodesk, 2012.

[15] Autodesk, "AutoLISP Developer's Guide", Autodesk, 2016.

[16] Y. Wang, J. Kandampully and H. M. Jia, "Tailoring customization services", *Journal of Service Management*, Vol. 24 Iss. 1, pp. 82 – 104, 2013.

[17] Autodesk, "Customization Guide", Autodesk, 2016.

[18] M. Rusiñol, A. Borràs and J. Lladós, "Relational indexing of vectorial primitives for symbol spotting in line-drawing images", *Pattern Recognition Letters*, 31(2010), 188–201, 2010.

[19] J. Zhou, P. E. D. Love, J. Matthews, B. Carey, C.P. Sing and D. J. Edwards , "Toward productivity improvement in electrical engineering documentation", *International Journal of Productivity and Performance Management*, Vol. 64 Iss 8 pp. 1024 – 1040, 2015.