Software Development of a Catalogue of Engineering Symbols as an Add-On Facility for use in CAD

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 – **Production of engineering drawings have evolved from manual to Computer Aided Design (CAD), primarily for the speed and accuracy presented by CAD. Unlike modelling and simulation packages that are equipped with building blocks, CAD packages are generally supplied with no pre-drawn symbols. From best practices in industrial engineering and working for companies that make use of CAD systems, evidently a lot of time is spent in repetitive tasks in the development of engineering drawings. An industrial engineering approach was adopted in this research where programming was used as a separate process from any specific application to an organization through the creation of a catalogue of almost 500 commonly used engineering symbols and storing them in a database, coupled with an add-on facility for AutoCAD for easy retrieval, aimed at avoiding these repetitive tasks, hence facilitating the rapid production of drawings to improve productivity and efficiency when using CAD systems.**

Keywords – **ASCII, AutoCAD, customization, symbols**

I. INTRODUCTION

Industrial engineering in operations and engineering management focusses on optimizing complex processes and systems to eliminate waste, reduce man-hours and machine time in order to enhance value and improve productivity and process efficiency [1, 2]. Such operations are hinged on the smooth operations of machinery which have to be maintained and in some cases modified and redesigned to cope with rapid changes in technology. Modifications or redesigns of machines require engineering designers to quickly develop solutions in order to avoid disruptions in operations and hence the invariable need for the quick generation of engineering drawings. Modern day machinery is driven by software that has also been rapidly changing. In order to optimize such operations, industrial engineers make use various software tools for drafting, modelling and simulations. The time required to produce the drawings and models can be significant hence the need to improve the process in order to reduce the man-hours required in the optimization of industrial operations. Modelling and simulation software packages such as STELLA [3], ASPEN [4] and ARENA [5] make use symbols to represent equipment in order to reduce the amount of time taken in modelling and eventual simulation and optimization of operations. However, Computer Aided Design and Drafting (CADD) software packages such as AutoCAD are supplied as base software but with no libraries of symbols to quicken the production of drawings in the same way.

Engineering drafting and design has gradually been transforming through the introduction of computers to enhance productivity and reduce workloads especially in updating and modifying designs. Owing to the frequency with which entities such as circles and lines are used, a lot of time can be saved and achieved by the creation of a library of symbols that can then be accessed in a CADD package [6]. Recent studies have revealed that most companies in Zimbabwe use CAD for producing drawings but little else explored for a host of other facilities that are offered by these software packages such as modelling, analysis, simulations or customizations to tailor make the packages for the companies' specific needs [7]. Many users of CAD packages are unaware of these facilities, presumably because of lack of knowledge or simply because they purchase these packages to achieve drafting purposes without worrying about what else they can use these packages for. AutoCAD is supplied with built-in American Standard Code for Information Interchange (ASCII) files that can be edited and customized to a company's specific requirements. Software developers for CAD packages normally supply libraries of symbols as separate software for specific applications, obviously the cost being the inhibiting factor. This research focused on the development of commonly used engineering symbols such as fasteners, bearings, electric motors etc. through programming and customization using the AutoCAD ASCII files. This was an innovative approach aimed at not only quickening the production of drawings by using symbols but it was also done in a generic way to allow users from engineering disciplines such as Electrical, Mechanical, Civil and Architectural Designs to make use of the add-on facility in AutoCAD without spending too much time on programming but simply configuring their systems to integrate with the add-on utility. This was also extended to the use of the same facility for parametric designs and the development of symbols of the same family that differ in dimensions, to support in-house development for companies that may find it unaffordable to buy Commercial Off-The-Shelf (COTS) packages which are generally expensive especially those separately supplied with libraries of symbols.

II. LITERATURE REVIEW

Until computers were widely available, products were designed manually on the drawing board by highly trained and skilled professionals. However, regardless of the skills, if changes were necessary, the drawings could be discarded and redrawn, consuming a lot of time in the process, adversely affecting the tenets of industrial engineering. The introduction of CAD made it possible to create drawings that could easily be modified, thus making it possible to create designs in a fraction of the time it took to produce the drawings manually [8]. AutoCAD AEC (Architectural, Engineering and Construction) was designed exclusively for the provision of AEC objects for the architectural, building and construction industries [9]. Other software packages such as IntelliCAD [10] and Trace [11] which include parametric designs are quite expensive and unaffordable to most small to medium enterprises in Zimbabwe, hence the need for development of customized and generic software as an add-on utility to existing CAD software. Programming and amending ASCII files that are provided with AutoCAD is not only affordable but is also convenient in saving time that may be required to adjust to COTS packages such as Trace and IntelliCAD. Although these packages do most of the routine and standard drawings, no two organizations are exactly the same in what they do and how they do it, hence the need to develop special but generic application software that meets the specific operational requirements of a given organization, business or discipline [12].

Blocks in AutoCAD can be used to build complex drawings from smaller details rather than redrawing commonly used symbols [7]. In AutoCAD, symbols can be can be drawn using normal procedures for CAD, stored in databases and retrieved and inserted into other drawings by defining the *x, y* and *z* coordinate location, scale and rotation factors. As such, any drawing can actually become a symbol and can be placed into another drawing [8]. Blocks minimize repetition and reduce drawing time especially when they are used to build complex drawings from smaller details rather than redrawing commonly used components [8, 11]. Virtually in any industrial engineering operation or business enterprise, blocks are standard components that are used to build machines or processes for modelling such as in ARENA, ASPEN or STELLA. Blocks in AutoCAD can be customized as libraries constructed to suit the user's needs [13]. These symbols are constructed once and can be used any number of times whenever they are required in a drawing [14]. Customizing symbol creation and management procedures involves selection of common symbols and drawing functions and implementing them as menu items or as automated drawing commands [8, 13]. A typical example is the customization of a program for a company that designs kitchen cabinets so that menus would prompt the user for the necessary information like the location on the wall where the cabinet will be placed, the cabinet's brand, model and dimensions [14].

There are various components in engineering made using the same industrial processes or procedures but the end products are available in different sizes [13]. Parametric programming generates drawings of components using variable parameters such as diameter and thread length in the case of a bolt. Bolts are however available in different sizes but in most cases they are produced using the same basic procedures, such that one is exactly similar to another. The programming comprises of a set of rules that instruct the computer on how to construct the component drawing. Parametric programming is also applicable whenever standard procedure is used for the design and where the designed component has standard dimensions. High productivity and short lead times require very rapid generation of drawings which can easily be achieved by parametric programming [15].

AutoLISP is a high level interface language for AutoCAD derived from the List Processing (LISP) programming language and allows users of AutoCAD to write macro-programs and functions that are well suited for graphics applications [16]. Many of the standard AutoCAD commands are AutoLISP driven applications. AutoLISP has the advantage of the flexibility of graphics manipulation and thus it is frequently used to code parametric programs in the design of similar components as well as for solid modelling [15, 17]. The AutoCAD interface can be modified by creating new or adjusting built-in ASCII text files to tailor-make AutoCAD to a company's specific needs and applications [18, 19].

In recent years, many organisations have been opting for COTS software for their operations owing to the low costs compared to customized in-house developed software and also the fact that COTS packages are developed by dedicated professional teams readily available to provide technical support, although at additional costs. However, industrial operations and machinery differ from one company to another, posing challenges to COTS software developers to try as much as possible to create software that satisfies the needs of many organizations. In that regard it becomes necessary to adjust or tailor-make the base software to suit the specific requirements of an organization. A wide variety of software tools and methods have been developed over the years to increase software quality and productivity [1]. These methods range from the traditional techniques such as the waterfall to the more modern and now frequently used techniques such as Agile Software Development (ASD) [20]. There are now even more approaches under ASD coupled with the rapid changes in technology and thus updates to these software packages. Whereas it is ideal to continually update software as and when COTS developers change or upgrade, the periodic updates in themselves do not necessarily ensure continual improvement [1]. For effectiveness, organizations benefit more by integrating new technology into an underlying process or existing base software. Although there were limitations, such integration proved to be effective when ARENA Simulation was used in conjunction with CPLEX to model and solve inventory and logistics problems through optimization [5].

Although 3D CAD modelling systems are now readily available for more realistic modelling of engineering components, some sectors still make use of 2D drawings, with an added need for transforming these into symbols that can be recognized by the more realistic solid modelers [21]. This is time consuming and costly to organizations hence the need for the creation of symbol libraries to avoid repetition of tasks. The scalability of such techniques has however been recommended for further research through analyzing other aspects such as electronic diagrams and mechanical schemes [6, 22], an aspect that this research deals with in the insertion of the created symbols into other drawings. Having established that a lot of time is spent in carrying out repetitive tasks because of the nature of most CAD base software, this research set out to answer questions on how engineering drawings can be generated rapidly in order to make the industrial engineering and other sectors productive and efficient.

III. METHODOLOGY

The research focused on the construction of commonly used engineering symbols and development of an add-on software facility for the AutoCAD environment through programming using ASCII and AutoLISP to achieve among other characteristics, the following; functionality, reliability, user friendliness, affordability and flexibility.

A. *Catalogue Development*

The catalogue of symbols was developed as a typical design problem from the realization of need through conceptualization to data description and finally the coded programs. To achieve the effective integration of the development cycle, an industrial engineering approach was taken and in this regard, the programming was treated as a separate process from any specific application to an organization although it was developed specific for the AutoCAD environment. The creation of symbols and their integration in a CAD system can be done using programming languages such as Borland C+ that are compatible with the CAD software, a macro-programming language such as AutoLISP or through the customization of the program menus. The researchers chose the latter option which was less time consuming and applied directly on the CAD environment. The ASCII text files that support customization in AutoCAD were modified or in some instances completely replaced by adding suitable macros to achieve the desired and specific applications for the engineering disciplines of Civil, Electrical and Mechanical as well as Architectural Designs as shown in Fig. 1. The same support files were used to store menu definitions and load AutoLISP programs or define line types and hatch patterns. Selection of the programming languages involved determination of how much time was required for coding, the expertise, flexibility in handling symbols and the efficiency of the programming techniques.

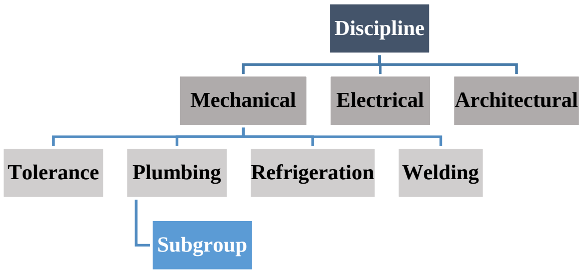

Fig. 1. Major disciplines and selected mechanical subcategory

The tasks for this research included the creation of; menu dialogue boxes, symbol searching modules, modules that do the actual insertion and those that manipulate the symbols. To customize the interface, three options were utilized, namely; use of script files, AutoLISP Programming as well as the ASCII text files for customizing menus. Script files were useful in automating commands but were limited especially where control was need to input parameters such as scale, rotation and insertion point on the symbol attributes. Whereas AutoLISP required extensive knowledge of LISP programming, the use of the ASCII text files required minimal or no programming experience but familiarity with their usage. The major advantage of this option was that these files are made specifically for information interchange hence the modification became relatively user friendly. The files also link with the program and thus there was no need to recreate the dialog and menu boxes. The ASCII option was thus chosen as it would take less time, hence less costly. The development cycle involved; 2D construction of the symbols in AutoCAD and storing them as blocks, classification of the symbols into relevant categories and subcategories as well as creating the menus through modification of the ASCII text files. Fig. 2 shows a screenshot of a typical block frequently used in architectural designs.

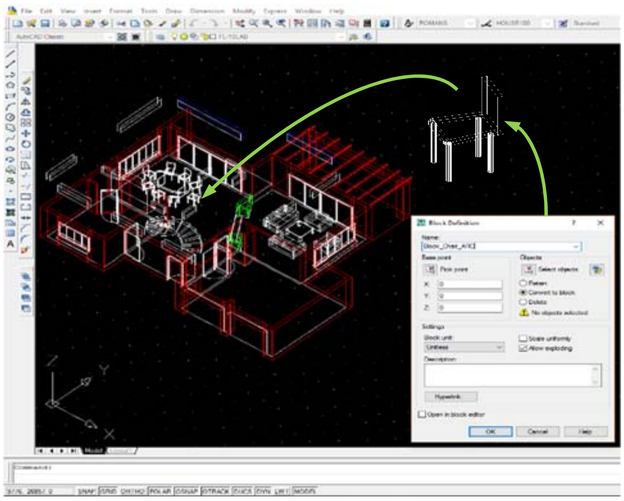

Fig. 2. AutoCAD screenshot of typical architectural block

B. Library Creation

The creation of the library of symbols (blocks) involved; classification of the symbols into categories under the 4 disciplines, creation of image slides (snapshot pictures of the blocks) and customization of menus. The on-screen and pull-down menus displayed on the AutoCAD screen are default menus supplied with the base AutoCAD software. These menus start from File, Edit through to Window and then Help as shown in the screenshot on Fig. 2. In the ASCII text files these are referred to as POP*n* menus were *n* is the position on the screen, e.g. POP2 refers to the Edit menu. The original ASCII text file for POP4 (Insert) appears as follows:

```
***POP4              Definition of the POP menu
[Insert]             Menu title
[Help!]'?            First item on the menu
[…………]
[Hyperlink]          Last item on the menu
```

IV. RESULTS

The ASCII text files for the "Insert" menu were modified to include the additional menu item of "Insert Symbol" by adding the following part of the extracted code at the end of the original ASCII file.

```
[Hyperlink]                          Last line on original menu
[………….]
 [-> Insert Symbol]
[->Mechanical]
   [Tolerance]$i=Tolerance $i=*
   [Plumbing]$i=Plumbing $i=*
   [Refrigeration]$i=Refrigeration $i=*
   [Welding]$i=Welding $i=*
[->Electrical]
   [Signal Systems]$i=Signal System $i=*
   [Lighting Outlet]$i=Lighting Outlet $i=*
[………….]
[->Architectural]
   [Doors]$i=Doors $i=*
   [Windows]$i=Windows $i=*
[………….]
```

After the pull down menus were all modified to include the additional "Insert Symbol" menu, symbol pictures derived from the 2D blocks were added for ease of retrieval. Subdirectories that contain the symbols were added to the AutoCAD batch file to enable these to be loaded at startup. To be available on the image plate files for ease of access, the Icon image plate file (***ICON) was modified such as for Tolerance Symbols, resulting in the image plate menu as shown in Fig. 3 on a typical AutoCAD model and drawing of a mining and industrial processing plant where such symbols were used.

```
**Tolerance                          Referencing name
[Tolerancing Symbols]                Title of Group
[Sym,Symetry]^C^CinsertSym           Tenth symbol on plate
[---------------]
```

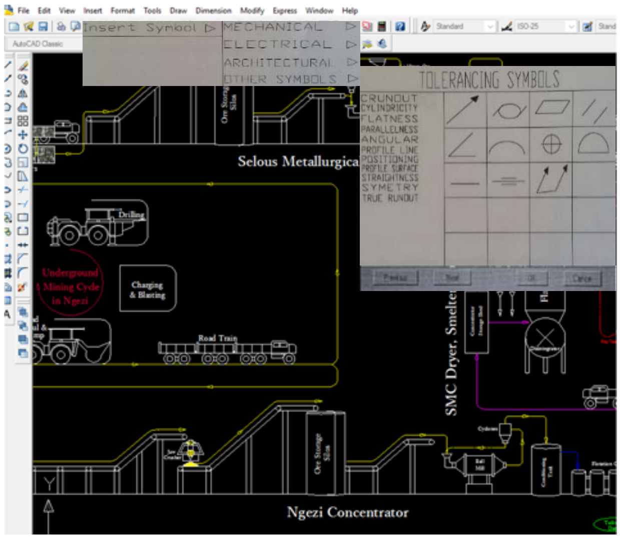

Fig. 3. Snapshot of the resulting image tile menu for tolerance symbols

V. DISCUSSION AND RECOMMENDATIONS

In addition to coding and customization of the Insert pull down menu and the image tile icons, an installation utility to aid the user to install the catalogue was also created. The main functions of the utility was to create a modified batch file to run AutoCAD with symbols. The costs incurred for developing this catalogue were mainly direct costs for the time taken to develop the three main categories of symbols and their contents, collection of drawing symbols from various sources, classification, customizing the pull-down menus, creation of the image tiles and insertion as well as preparation of the user manual as summarized in Table I.

TABLE I
ESTIMATED COST FOR THE SOFTWARE ADD-ON

Activity/Item	Unit Cost/USD	Subtotal/USD
Collection of Symbols	15 hrs @ $10/hr	$150
Drawing Symbols and Slides	40 hrs @ $10/hr	$400
Customization	10 hrs @ $10/hr	$100
Preparation of Manuals	10 pps @ $2/pp	$20
Estimated Indirect Costs		$100
Grand Total		**$770**

The indirect costs were mainly contingent costs that included support staff and other associated overheads. Even though the objectives of developing the in-house add-on facility were largely achieved, there were areas that still required some further refinements and improvements such as to develop the suite of symbols to include those that may be used for 3-D modelling and simulation as in ARENA, STELLA and ASPEN. Although the use of parametric drawings and designs using AutoLISP in this research were limited to the generation of some of the symbols, it may be prudent to add this aspect to allow for the rapid generation of symbols within symbols such as gears and fasteners. For the future versions of the add-on software, there should be a provision for the user to add their own symbols to the library but it must be done in a careful manner to avoid disabling the functionality of the add-on facility. This provision should also allow the user to modify symbols to suit their specific needs. The manual can be expanded to include a basic tutorial for operating AutoCAD as some users may be new to the package. Autodesk has been upgrading their software to new versions at a faster pace than it were a few years ago. The add-on facility must correspondingly be updated to match the new releases of AutoCAD otherwise the add-on facility may lose its functionality as the versions are upgraded. Although this add-on facility was developed in a generic form to cater for users in the 3 main disciplines of engineering and architectural designs, this is limited for use with AutoCAD, hence a need to further the research and look at ways of integrating this with other CAD package.

Further programming and modifications of the ASCII text files may be necessary to include attributes of some of the symbols, such as voltage for battery symbols to enable the user to specify these before inserting, thus ensuring that the right symbol with the right rating has been picked. Close to 500 commonly used engineering and architectural symbols were created during the research. However there are more and new symbols adopted as technology changes, as well as those other disciplines that were not catered for in this project could have symbols of their own. As the add-on facility is progressively updated, it is recommended that more and more symbols can be added to cover a wider spectrum of the categories and subcategories of the disciplines.

VI. CONCLUSION

Drawing symbols that cover the three major disciplines of engineering and architectural designs were collected and classified into different categories and subcategories depending on the functions and close to 500 of these were created in 2D as blocks and stored in a suite of engineering symbols for use in AutoCAD. The base software package was modified through ASCII and AutoLISP programming and customized to enable access to the suite. An installation utility that loads the symbols automatically at startup in an existing AutoCAD environment was also created and these can be accessed and inserted in drawings by specifying the scale, rotation and coordinates of the insertion point. The cost of developing the add-on facility was estimated to be USD 770, a reasonable and affordable cost for in-house development compared to COTS software.

An industrial engineering approach was taken in this research in that the programming was treated as a separate process from any specific application to an organization although it was developed specific for the AutoCAD environment, hence the need for further research to make the add-on facility adaptable to any CAD software. Industrial and engineering companies can benefit from this add-on facility through the rapid production of standard drawings and increase in productivity.

REFERENCES

[1] 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.

[2] 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.

[3] M. Biss, “Testing Overload Levels of Audio Amplifier using STELLA”, *Procedia Computer Science* 43(2015), 95-100, 2015.

[4] M. Sajjad and M. G. Rasul, “Simulation and Optimization of Solar Desalination Plant Using ASPEN Plus Simulation Software, *Procedia Engineering*, 105(2015), 739-750, 2015.

[5] M. Vamanan, Q. Wang, R. Batta and R. J. Szczerba, “Integration of COTS software products ARENA & CPLEX for an inventory/logistics problem”, *Computers & Operations Research*, 31(2004), 533–547, 2004.

[6] A. G. Yuferov, “Library of graphic symbols for power equipment in the scalable vector graphics format”, *Nuclear Energy and Technology*, 2(2016), 64–69, 2016.

[7] W. R. Nyemba and J. Lambu, “A CAD system for the automatic generation of bills of materials for a motor vehicle manufacturing company”, *Proceedings of the Zimbabwe Institution of Engineers*, Vol. 4 Issue 1, pp. 32-39, 2006.

[8] M. Fox, W. B. Holtz and D. Jordani, “Low Cost CADD: A Decision Guide”, *PC Magazine*, Vol. 6, December 1987.

[9] Autodesk Inc., “AutoCAD 2002”, Autodesk, 2001.

[10] IntelliCAD, “IntelliCAD Technology Consortium Announces Release of IntelliCAD® 8”, *ITC Press*, 2014.

[11] Trace Software, “Industrial Process Schematic Design”, Available: http://www.trace-software.com/, Accessed: 12 June 2016.

[12] J. Michniewicz, G. Reinhart and S. Boschert, “CAD-based automated assembly planning for variable products in modular production systems”, *Procedia CIRP*, 44(2016), 44-49, 2016.

[13] Autodesk, “AutoCAD 2013 User’s Guide”, Autodesk, 2012

[14] T. Guo, H. Zhang and Y. Wen, “An improved example-driven symbol recognition approach in engineering drawings”, *Computers & Graphics* 36(2012), 835–845, 2012.

[15] Autodesk, “Parametric Modelling and Product Design Features”, Available: http://www.autodesk.com/, Accessed: 13 June 2016.

[16] Autodesk, “AutoLISP Developer’s Guide”, Autodesk, 2016.

[17] R. Marschallinger, C. Jandrisevits and F. Zobl, “A visual LISP program for voxelizing AutoCAD solid models”, *Computers & Geosciences* 74(2015), 110–120, 2015.

[18] Autodesk, “Customization Guide”, Autodesk, 2016

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

[20] G. Papadopoulos, “Moving from traditional to agile software development methodologies also on large, distributed projects”, Procedia - Social and Behavioral Sciences, 175, (2015), 455 – 463, 2015.

[21] H. J. Hwang, S. Han and Y. D. Kim, “Recognition of design symbols from mid-ship drawings”, *Ocean Engineering*, 32 (2005), 1968–1981, 2005.

[22] 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.