Advanced Engineering Informatics 26 (2012) 231–250

Contents lists available at SciVerse ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Adaptable methodology for automation application development

Christian van der Velden [a], Cees Bil [a,*], Xinghuo Xu [b]

[a] School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, Australia
[b] School of Electrical and Computer Engineering, RMIT University, Melbourne, Australia

A R T I C L E I N F O

Article history:
Available online 10 March 2012

Keywords:
Knowledge-based engineering
Design automation
CommonKADS
MOKA
Productivity enhancement

A B S T R A C T

An adaptable methodology for automation application development (AMAAD) is introduced. This development methodology is based on the key concept that design automation (DA) applications are a subset of higher level knowledge-based engineering (KBE) applications, and thus can be developed using a subset of KBE methods. The proposed methodology is largely built upon two existing KBE methodologies: CommonKADS and MOKA, which have become popular models for automating engineering processes. The proposed extension of these methods introduces flexibility to tailor the process for producing automation software to the specific needs of the problem through the specification of a number of attributes. These attributes are linked to subtasks in the key lifecycle phases of application development. This proposed methodology provides a link between KBE and DA applications and provides structure to the application development process. A software tool was written to facilitate the process of identifying the capability needs of an automated solution, and providing detail of the tasks to be followed for its development.

1. Introduction

Knowledge-based engineering (KBE) is the field of engineering concerned with capture, management and utilisation of engineering knowledge relating to processes in a product development lifecycle, for implementation in software systems that automate engineering processes [1–4]. This knowledge is often unique to the product manufacturer, based on previous development experience. KBE technology provides the capability to:

- automate processes in a product development lifecycle, leading to a reduction in time and cost,
- ensure consistent quality of outputs from an engineering process,
- verify designs against standards,
- capture engineering knowledge for later reuse,
- retain knowledge of domain experts,
- provide structure to development processes.

Traditionally, KBE was closely associated with geometric modelling in CAD systems; allowing geometry to be rapidly created using sets of rules describing the steps in the process. However, modern KBE application capabilities extend to other areas of the product development process including design, analysis, manufacture and ongoing support. Two seemingly different interpretations of process automation are generally adopted by academia and industry.

The academic view of KBE considers the process for building knowledge based systems to be a comprehensive modelling task, involving much more than simply writing software to automate a process. Rather, a detailed study of organisational practices should be conducted, and detailed models of numerous facets of domain and product knowledge be constructed before system design even begins. Typically, the knowledge based view of process automation aims to remain generic as far into the design process as possible, with problem specific methods and data defined at the latest levels. This extends maximum flexibility to KBE applications, providing more opportunities for reuse, and improves processes for modification and upgrade of methods and data. The nature of this modelling approach is such that interrelationships between knowledge objects and entities are preserved and integrated solutions are produced.

Industry often takes a more pragmatic view of this development process, instead focusing on a specific need and developing a system to meet that need with tangible benefits in terms of reduced lead time and cost. In such cases, the complete development process as specified by KBE models may not be practical to implement due to excessive time requirements. It can be argued, however, that this type of automation is not truly KBE, and can be better described as design automation (DA). Industrial applications of DA generally involve coding and deploying functional applications to address problems well within the timeframe of otherwise completing the original tasks manually. These applications are typically purpose-

* Corresponding author. Tel.: +61 399256176; fax: +61 399256108.
E-mail address: bil@rmit.edu.au (C. Bil).

1474-0346/$ - see front matter
doi:10.1016/j.aei.2012.02.007

oriented, having limited scope and lifetime. From an industrial viewpoint, significant cost savings can be made with the ability to produce automation applications quickly. The differences between the academic and industrial perspectives of automation can be seen as a trade-off between application completeness against economic viability. The former offers solutions that provide full automation of engineering tasks, while the latter requires a sound business case that demonstrates tangible savings often with aggressive scheduling constraints, thus compromised solutions are often the result.

Various differences between KBE and automation are discussed in Section 2 leading to the motivation behind the development of the AMAAD methodology. Section 3 discusses a number of existing KBE methodologies on which AMAAD is based. Section 4 gives a detailed description of the AMAAD methodology, how it structured and applied. Section 5 gives a brief introduction to the application of AMAAD to the problem of routing of wire harnesses and piping through a complex structure, a typical task in the aerospace industry, with a discussion on strengths and weaknesses as well as some directions for further development and investigation. Section 6 gives conclusions on the finding of the project.

2. Knowledge-based engineering versus design automation

KBE applications provide more flexibility and adaptability than automation applications that simply link two dissimilar systems or datasets, or automate a simple process. Traditional approaches to DA are described as "*piecewise automation or a tightly coupled, hard-coded procedure-based programming*" [1]. Accordingly, when new methods are to be incorporated, significant changes in code are required. DA applications lack effective mechanisms for communicating with other systems, and flexibility to apply to new situations.

KBE systems are described as exhibiting five main characteristics including: dynamic, generic, generative, high-level, and demand driven [1]. These characteristics refer to the capability of resulting KBE applications to:

- reconfigure rules and outputs based on new inputs,
- handle new known and unknown problems,
- derive new rules automatically from old rules based on input changes,
- provide high level commands that invoke a number of subprocesses,
- intelligently control rule sequencing and execution.

While there are several obvious and significant benefits over DA systems, the implementation of KBE systems to a high level of detail is a very complex task, and not practical for companies without extensive resources. For these companies, significant process improvements can be made with the introduction of DA technologies, however, it is acknowledged that there are several shortcomings of these methods; primarily among a lack of formal structure to the development process.

KBE and DA applications differ in scope. KBE applications are viewed as integrated systems solutions, with dynamic links to the governing knowledge base which can be reconfigured as new information becomes available and requirements evolve over the product lifecycle. Conversely, DA applications tend to be stand - alone programs that automate engineering tasks, but lack the dynamic nature of KBE applications. Rules tend to be hard-coded, requiring updates to source code when changes in data and requirements occur. However, due to the relative level of detail of DA applications, development and deployment timescales tend to be much shorter, allowing working solutions to problems to be delivered to engineers quickly for use on projects. Table 1 summarises some of the differentiating characteristics typical of KBE and DA techniques.

There are conflicting views on the "best" way to implement automated solutions. DA solutions provide a quick fix for problems in a relatively short time, but applications are typically developed independently with little or no communication between them. While in the short term they may be useful to end users on an individual basis, in the long term many of these solutions may be required for the development of the same product, using the same datasets. The management of the large amounts of data then becomes a significant issue. In a true KBE system, the various processes would be typically linked in a larger knowledge model.

2.1. Applications and capabilities

The use of KBE in the aerospace industry is focussed on implementing knowledge of product development processes in individual software applications to automate engineering tasks. As such, from a DA, as opposed to a KBE, viewpoint, it is important to recognise that not all processes are suitable for automation. Some tasks, especially those requiring tacit human judgement, will always require some level of user input, and the effort required to automate such processes often does not outweigh the benefit gained by automated capability. For automation of such processes, higher level knowledge based systems should be implemented, requiring a higher level of development effort and time. Typical processes which lend themselves well to automation generally exhibit one or more of the following characteristics [11]:

- low level, repetitive, and/or highly manual tasks,
- integration of tools and datasets (e.g. CAD/CAE/CAM),
- automated documenting and report generation,
- simplification and/or standardisation of more complex processes,
- generation of manufacturing data and tooling design.

Two main implementations of automated solutions are commonly used in industry. The first involves a formally identified task with well defined requirements, developed in multidisciplinary teams which may include subject matter engineers, software engineers and programmers. There must be a business case for developing such solutions, i.e. provide a positive return on investment (ROI). Eq. (1) shows that the ROI is calculated from the ratio of number hours required to perform the task completely manually, versus the number of hours developing the solution, including training, and completing the task automatically [10]:

$$ROI = \frac{n \times t_M}{t_D + n \times t_A} \tag{1}$$

where ROI: return on investment; n: number of instances of task; t_{MB}: time to complete one instance of task manually; t_{DB}: time to develop automated solution; t_{AB}: time to complete one instance of task automatically using automation tool.

As this type of solution is rolled out and made available to engineers, usage is logged for comparison with performance forecasts, and feedback for future improvements to the application or development of new applications. This also assists in marketing automation methodologies to management and customers. Aside from the tangible cost and scheduling benefits, automation of processes also provides intangible benefits including [11]:

- **Consistency: Dedicated** tools with standardised inputs and outputs can provide greater consistency.
- **Complexity: Simplification** or standardisation of complex processes, minimising scope for human error.