Journal of Computational Design and Engineering Vol. 1, No. 3 (2014) 161~172
www.jcde.org

A comparison of three design tree based search algorithms for the detection of engineering parts constructed with CATIA V5 in large databases

Robin Roj[1,*]

[1] University of Wuppertal (FB D Mechanical Engineering, Mechanical Engineering Informatics, D 42097 Wuppertal, Germany)

(Manuscript Received January 26, 2014; Revised April 24, 2014; Accepted April 24, 2014)

Abstract

This paper presents three different search engines for the detection of CAD-parts in large databases. The analysis of the contained information is performed by the export of the data that is stored in the structure trees of the CAD-models. A preparation program generates one XML-file for every model, which in addition to including the data of the structure tree, also owns certain physical properties of each part. The first search engine is specializes in the discovery of standard parts, like screws or washers. The second program uses certain user input as search parameters, and therefore has the ability to perform personalized queries. The third one compares one given reference part with all parts in the database, and locates files that are identical, or similar to, the reference part. All approaches run automatically, and have the analysis of the structure tree in common. Files constructed with CATIA V5, and search engines written with Python have been used for the implementation. The paper also includes a short comparison of the advantages and disadvantages of each program, as well as a performance test.

Keywords: CAD; CATIA V5; Classification; Database; Dat mining; Design tree; Feature recognition; Knowledge Discovery; Python; Search engine

1. Introduction

Contemporary construction techniques of engineering parts are heavily influenced by the usage of computer-aided methods for the virtual design of new products, as well as for the management of whole projects. The appropriate software is available for the different demands, like e.g., the conception and technical draft of single parts, or the administration and coordination of several other engineering tasks, and can be summarized as Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE), respectively. The shift from manually drawn sketches and sophisticated manufacturing of prototypes in the development phase of new components to Virtual Product Development (VPD) supported by simulations or computations has yielded many advantages, in terms of cost effectiveness and quality improvement.

But new challenges and tasks have arisen with the execution of computer-aided technology. One major difficulty is the handling of large amounts of data that are produced during the usage of software belonging to Product Lifecycle Management (PLM). Just a few examples are specialized files for virtual models, technical drawings, FEM-calculations and assemblies, or additional data, like tables, images, presentations, and even videos. A common strategy in large companies like car manufacturers, who produce a lot of the mentioned information, is storage in large databases with company-wide access.

An important question is the setup and the structure of such a database, depending on the desired objectives. These intentions could be e.g., good documentation of the accomplished work or might even include all possibilities of comfortable reaccess to the stored information. Because of the many imaginable styles of the warehousing of data in databases, like alphabetical or chronological order, as well as sorting depending on the different departments or branches of a company, it can be a challenging task to regain once stored information.

For this research area, the keywords Knowledge Discovery in Databases (KDD) and Data Mining are introduced. According to Fayyad, Piatetsky-Shapiro and Smyth KDD, can be defined as a procedure [1]: "*KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.*"

The same authors describe the approach of KDD, as follows [2]: "*KDD focuses on the overall process of knowledge discovery from data, including how the data are stored and accessed, how algorithms can be scaled to massive datasets and still run effectively, how results can be interpreted and visualized, and how the overall man-machine interaction can*

*Corresponding author. Tel.: +49-202-439-2090, Fax.: +49-202-439-2091
E-mail address: r.roj@uni-wuppertal.de
© Society of CAD/CAM Engineers & Techno-Press

doi: 10.7315/JCDE.2014.016

usefully be modeled and supported."

Frawley, Piatetsky-Shapiro and Matheus consider the three most important aims of KDD to be summarization, discrimination and comparison, referring to a clear distinction and categorization of the properties of extracted data [3].

Data Mining describes a similar strategy and the etymology developed according to Petersohn from colloquial language for the exploitation or extraction of rare materials, compared with valuable information from large amounts of data [4]. A good overview is provided by Gorunescu [5]. Hilderman and Hamilton describe an evaluation of the extracted information for the measurement of interestingness. They consider classification, association, clustering and correlation as the four most important techniques for the extraction of data [6].

This publication presents a specific case of data extraction. For the implementation of the algorithms, the CAD program CATIA V5, and the programming language Python have been chosen. Due to the fact that Python is connectable with CATIA via the COM-interface both programs are predestinated for the creation of macros, which are able to automate certain steps that are usually done by hand. The examination of a large database filled with engineering parts constructed with CATIA is demonstrated. As already mentioned the aim is a certain kind of classification of random components that should work as automatically and independent as possible.

At this point, the constitution of CAD models generated by most CAD systems, as well as CATIA should be explained briefly. Usually, the constructor builds the virtual model of the engineering part by the usage of several functions, like e.g., extrusions, rotations, drillings, roundings, or chamfers, to gain the intended shape. These properties of each part are called features. According to Vajna et al. [7], features can not only be labeled as geometrical elements, but also as relevant informational elements like relations and constraints.

The set of all features, and the unambiguous determination of every feature a model consists of, forms the precise definition of the part; and the combination of both leads to a certain singularity. All common CAD systems save these features in the created files, and also in the so called construction tree or design tree, where the user is able to comprehend the contained specifications. According to Kornprobst, the structure tree illustrates all the construction steps, which lead to explicit geometry or rules in a chronological order [8].

In Sections 3 and 4, three search algorithms are introduced, which examine the structure trees of given CAD models. The information contained in the structure trees are outsourced from CATIA, and stored in the form of XML-files for easier access, and subsequent classification. In Section 5, the advantages and disadvantages of the three search algorithms are compared, and the cases they are more or less appropriate for are determined.

2. State of the art

For the setting described in Section 1, many scientific and industrial concepts already exist. Here it should focus on engineering applications, and thus databases that are filled with CAD and CAE files, respectively. Approaches dealing with the general structure of such information are presented by Ester et al. [9], who focus on spatial databases; as well as Haffey and Duffy [10], who connect the topic with design issues. The dissertation of Angkasith concentrates on modular design [11]. In particular, the management of engineering products, which are manufactured by several suppliers and only mounted by the principal, is quite ambitious, regarding the administration of the generated data, and the coordination of every single working step. For such cases, a separation of the final product into modules might be time and cost reducing.

The representation of knowledge, and the interaction of the single elements with each other, can be visualized e.g., by directed graphs. Also, the above mentioned structure tree of CATIA is considered as a graph, and therefore outsourced into the XML-format, which is able to illustrate hierarchical structures. An example of the usage of graphs for the description of complex circumstances is presented by Kizu et al. [12]. They show a method for CAD Data Mining, and the detection of two-dimensional objects.

For engineering applications, not only the virtual construction of new products is important, but also a well elaborated production plan. Consequently, a strict separation of CAD features and manufacturing features take place, and has to be taken into account during the automatic feature recognition. A comprehensive review of Data Mining in manufacturing is given by Harding et al. [13].

Babic, Nesic and Miljkovic list the three main problems of Automated Feature Recognition (AFR) as 1. Extraction of the geometric primitives from a CAD model, 2. Defining a suitable part representation for form feature identification, and 3. Feature pattern matching/recognition [14]. In another review by Iyer et al. [15], the following techniques for the detection of shapes are structured in six categories: 1. Global feature-based techniques, 2. Manufacturing feature recognition-based techniques, 3. Graph-based techniques, 4. Histogram-based techniques, 5. Product information-based techniques, and 6. 3D object recognition-based techniques.

Two vivid examples of the automated recognition of three-dimensional objects are suggested by Min and Bowyer [16], who detect edges and reconstruct surfaces by image segmentation; and Cucchiara et al. [17], who use visual constraint graphs for an analysis of spatial components. Relational graphs are also used by Flynn and Jain [18]. They connect their topic with the storage of gained information in a database, which might later be used as a basis for manufacturing. While their publication deals with a library of proprietary CAD files, Cybenko, Bhasin and Cohen plan a global system for the representation, and in particular the reuse of once detected shapes [19]. By applying the examination techniques already in the design phase, and the usage of voxel-

Download English Version:

https://daneshyari.com/en/article/442286

Download Persian Version:

https://daneshyari.com/article/442286

Daneshyari.com