Available online at www.sciencedirect.com

ScienceDirect

Computer-Aided Design 40 (2008) 828–837

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Tracking product specification dependencies in collaborative design for conflict management

M.Z. Ouertani[a,*], L. Gzara[b]

[a] CRAN (Research Centre for Automatic Control), CNRS UMR 7039, Nancy University, Faculty of Sciences and Techniques, BP239, Vandoeuvre-les-Nancy, 54506, France

[b] G-Scop, CNRS FRE3028, INP-Grenoble, France

Received 15 September 2006; accepted 3 July 2007

Abstract

The main difficulty associated with a collaborative design process is understanding the product data exchanged during the design. Efficient and effective coordination of design activities relies on a thorough understanding of the dependencies between shared product specifications throughout the entire development cycle. This paper explores the linkages between the design process features and product specification dependencies, and suggests ways of identifying and managing specification dependencies to improve collaborative process performance. Using a UML (Unified Modeling Language) specification, we propose a process traceability tool to track the design process in an ongoing manner. Based on the information captured, the dependencies between specifications involved in the tracked process are identified and inserted in a dependency network, which is maintained throughout the design process. A set of mechanisms is then proposed to qualify the identified dependencies. Extracting and qualifying specification dependencies could be useful in many design situations; for example, during an engineering change management process to assess impacts and study change feasibility, or during a conflict management process to assist designers in resolving conflicts and maintaining the coherence of the design process (knowing that change management is a tool to conduct conflict management). Special attention is paid to the conflict management process. By means of a case study, we show how the solution we propose can assist designers during the conflict management process.

Keywords: Collaborative process; Design process traceability; Specification dependencies; Conflict resolution; Change management

1. Introduction

Although in most design processes, coordination entails clear communication between designers, the real reason for this coordination is not for communication but for resolving dependencies between product specifications [1]. Design is constraint oriented, and comprises many interdependent parts. A change in one part may have consequences for another part, and designers cannot always oversee these interdependencies and consequences.

Product specifications are not always trivial and explicit. Current Product Data Technology (PDT) tools are not able to extract these dependencies from informal and textual descriptions, and hence the dependencies cannot be revealed by the currently available PDT tools.

Several researchers have investigated specification dependencies, for example: Eppinger et al. [2], Kusiak and Wang [3], Dong and Agogino [4], Wang and Jin [1], Browning [5] and Yassine and Braha [6]. Most have proposed a representation of specification dependencies (with a DSM[1] matrix, a network or a set of patterns) but none has shown how to obtain these representations or proposed mechanisms to identify dependencies. Moreover, these studies have addressed the specifications involved in a design process that *has already been carried out*, whereas the usefulness of specification dependencies knowledge is primarily *during* the design process, to help designers to perform their activities and resolve interdependency problems. In addition to this reported work, all studies reported to date have only investigated the case of two dependent design activities (an upstream activity feeds specifications to a downstream activity) belonging to the same decomposition level of

* Corresponding author. Tel.: +33 3 83 68 44 30; fax: +33 3 83 68 44 59.
E-mail address: Mohamed-Zied.Ouertani@cran.uhp-nancy.fr (M.Z. Ouertani).

[1] Design Structure Matrix.

0010-4485/$ - see front matter
doi:10.1016/j.cad.2007.07.002

the design process. For a complete identification of specification dependencies, it is necessary to consider the dependencies between activities carried out at various decomposition levels of a design process.

In terms of dependency qualification, some researchers have provided interesting proposals on this issue. We particularly note the framework developed by Krishnan et al. [7] to measure dependency based on two dimensions: upstream specification *evolution* and downstream specification *sensitivity*. We also cite the work of Kusiak and Wang [3] dealing with *qualitative dependency* (the direction of the change of a product specification that is affected by another) and *quantitative dependency* (the rate of change of a product specification that is affected by another). However, these dependency qualifications and quantifications do not assist in identifying whether the dependency is strong and should be kept, or whether it can be removed. We should highlight that all of the previous studies have assumed the importance of qualifying product specification dependencies but none of them has proposed mechanisms to support these dependencies.

We have therefore developed a solution called DEPNET,[2] which explicitly captures product specification dependencies, inserts them in a dependency network that is maintained throughout the design process, and assists designers in resolving dependencies during the design process, particularly when design conflicts occur or when engineering changes are requested.

The DEPNET solution differs from the previous studies [1–7] in three major aspects, as follows.

- It proposes a method to identify specification dependencies and defines concepts to qualify the discovered dependencies.
- It takes into account the predefined specifications as well as the emerging specifications resulting from non-planned design activities.
- It considers the design process in a more realistic way and seeks to identify product specification dependencies among sets of dependent activities belonging to various decomposition levels.

In addition to the introduction, this paper consists of four more sections. Section 2 illustrates the problem definition, using the example of a turbocharger design problem. Section 3 focuses on the DEPNET solution: it describes the dependency network constructs and shows how to build this network. Section 4 illustrates, by means of a case study, the use of the DEPNET solution during conflict management, while Section 5 provides conclusions and future prospects.

2. Example — A turbocharger design process

The mechanical concept of a turbocharger revolves around three main parts: a *turbine wheel*, driven by the exhaust gas from a pump, most often an internal combustion engine, which spins the second part; an *impeller*, whose function is to force more air into the pump's intake supply; and a *centre hub rotating assembly* (CHRA), which contains bearings, an oil circuit, a cooling system and a shaft that directly connects the turbine and impeller.

The design process of a turbocharger is composed of four main concurrent sub-processes: the turbine wheel design sub-process, the impeller design sub-process, the CHRA design sub-process, and the centre housing assembly customisation (i.e., matching the impeller housing and turbine housing to the vehicle engine). The designers have then to exchange preliminary product specifications to enable the process to move forward.

At the beginning of the turbocharger design process, the concerned parties each have a set of data at their disposal as well as a set of requirements to follow, such as:

- the vehicle/application specifications: vehicle or equipment manufacturer, gearbox type and maximum weight of the vehicle,
- the engine specifications: engine type, configuration, displacement and 3D CAD drawings,
- the engine performances: power, torque, engine speed and air flow, and
- the turbocharger specifications: regulation type, actuator type, engine turbocharger position, turbocharger weight and specific speed limits.

According to these specifications, the *impeller designer* starts the planned activity (to define the impeller part). The designer has to define the impeller attributes, which include: wheel cast-material, wheel cast-process, expected compressor inlet temperature and compressor outlet temperature. Once these attributes are fully defined, the impeller designer defines the exducer and inducer diameters of the compressor wheel. The impeller 3D CAD drawing is then created. The final task in the impeller design sub-process is defining the impeller housing by calculating the parameters' 'trim'[3] and A/R.[4] Once these attributes are calculated, the designer is able to complete the 3D CAD drawing of the impeller part.

Based on the customer specifications, the turbocharger specifications and the impeller-defined attributes, the *turbine designer* commences the planned activity (to define the turbine wheel) at the same time as the impeller designer commences the planned activity. First, the interdependent parameters, namely the wheel, nozzle ring and insert ring materials, the maximum limit of the turbine inlet temperature, and the inlet/outlet turbine pressure are defined in such a way as to achieve the target turbocharger performance. Once these parameters are defined, the turbine designer can begin to define the wheel dimensions and create the 3D CAD drawing of the turbine wheel. Defining the wheel dimensions involves calculating the exducer and

[2] DEPNET — product specification *DEP*endencies *NET*work identification and qualification.

[3] The 'trim' attribute, which is an area ratio used to describe both turbine and compressor wheels, is calculated using the inducer and exducer diameters.

[4] A/R describes a geometric characteristic of all compressor and turbine housings. It is defined as the inlet cross-sectional area divided by the radius from the turbo centreline to the centroid of that area.