Knowledge-Based Systems 70 (2014) 256–267

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Knowledge evaluation in product lifecycle design and support

Yang Xu [a,*], Alain Bernard [b], Nicolas Perry [c], Jing Xu [d], Shigeo Sugimoto [e]

[a] *Department of Information Management, Peking University, 100871 Beijing, China*
[b] *IRCCyN, Ecole Centrale de Nantes, 44321 Nantes, France*
[c] *Arts et Métiers ParisTech, I2M, UMR 5295, F-33400 Talence, France*
[d] *College of Mechanical Engineering, Yangzhou University, 225009 Yangzhou, China*
[e] *Research Center for Knowledge Communities, University of Tsukuba, Ibaraki 305-8550, Japan*

ARTICLE INFO

Article history:
Received 10 January 2014
Received in revised form 22 June 2014
Accepted 9 July 2014
Available online 18 July 2014

Keywords:
Knowledge management
Knowledge value
Product design
Life cycle management
Evaluation

ABSTRACT

Enterprises are focusing more and more on knowledge issues for global product development. This paper describes knowledge evolution processes in product development activities and proposes a knowledge evaluation method in product lifecycle design. The paper also theoretically analyzes the evaluation model and illustrates how knowledge values can be assessed by case study. The case study shows how knowledge values calculated by the model can provide suggestions about which knowledge to choose and what to do next. The knowledge evaluation model serves as a useful tool for managing knowledge in product lifecycle design and support.

1. Introduction

The current commercial environment necessitates that enterprise to adapt to the requirement of more innovation, fewer errors, less time-to-market, lower manufacturing cost, better operational performance and better cooperation among partners [4]. In such situations, more and more enterprises consider their production processes as knowledge management (KM) processes, and they are paying attention to the crucial competence: knowledge [25,32,21]. Meanwhile, the whole lifecycle plays an important role in production activities. So product lifecycle management (PLM), from the initial conception to the end of life, is a strategic approach in production management [26].

A variety of intelligent solutions have been proposed in knowledge management concerning product development. Karacapilidis [19]proposed a computerized knowledge management system for the collaborative development of a manufacturing strategy. Their system supports collaborative strategy development by integrating a domain-specific modeling formalism based on the resource view of the firm, an associated structured dialogue scheme, an argumentation-enabling mechanism, and an efficient algorithm for the evaluation of alternatives. He et al. [16] proposed a unified product structure management model to integrate product structure information and enterprise business processes and to ensure people of various disciplines can access product information throughout the entire product lifecycle. Hung et al. [18] have developed a novel framework supported by a knowledge-based database to support product design planning, considering quality function deployment and design structure matrix. Chen [12] has presented a five-step approach using knowledge integration and sharing mechanism for collaborative modeling product design and process development. It can satisfy participants' demands for product knowledge, increase product development capability, reduce product development cycle time and cost, and ultimately increase product marketability. Gunendran and Young [15] have conducted surveys on how to organize manufacturing best practice knowledge in product development, and they have explored a system design tool to model the relationship between knowledge and product information so as to reuse system design models. Chang et al. [9] have studied organizational knowledge structure in the context of new product development (NPD) and illustrated that one must possess enough working experience within product development process to have the skills to accomplish cross-functional knowledge conversion. Al-Ashaab et al. [2] have implemented the knowledge-based environment framework KBE-ProVal (Knowledge-Based Environment to Support Product Design Validation) to support product design validation. Akasaka et al. [1] have extend product design to Product-Service-System (PSS) and proposed a knowledge-based PSS design support method.

Those results show that when production is tightly linked with knowledge, product development solutions do not focus just on the

* Corresponding author.
E-mail address: yang.xu@pku.edu.cn (Y. Xu).

http://dx.doi.org/10.1016/j.knosys.2014.07.003
0950-7051/

"product" but extend to knowledge management. As a result, product designers should not only combine enterprise business processes with product development processes but also integrate all functional elements which could be identified, especially knowledge. Effective models are expected to be a unified platform for creation, sharing and application of knowledge that is related to product and production activities [23]. In other words, product design should consider all knowledge in all stages of a product lifecycle processed by all participants, linking to enterprise lifecycle management, technology lifecycle management and associated to a global knowledge lifecycle management.

However, knowledge evaluation is a topic within KM that is not well studied especially when integrated with product development. Evaluation is crucial for knowledge management in both research and practice, however, the intangibility of knowledge make evaluation very complex. Only by developing a standardized and quantitative approach can we establish a method of knowledge evaluation that can be applied in practice. Xu and Bernard [32] have proposed a basic knowledge quantification approach which evaluates how knowledge can make changes in product state evolution. Based on the idea and approach, this paper addresses the problem of knowledge evaluation for further application, discusses how knowledge can improve product lifecycle design process, validate the efficiency of knowledge evaluation process, determine the optimized sequence of in knowledge acquisition, and provide enterprises with a global view on product design.

2. Knowledge evaluation modeling

2.1. Product development process description

In a product development process, a product may be considered to start from its initial state and arrive to a required state (final state), and a task T is supposed to be accomplished to realize this product evolution from that initial state P_0 to the final state P_n. For example, to produce a car (product), here is one step of the product development process: the car is to change from version 1.0 (initial state) to version 2.0 (final state), and a task T can bridge the gap between these two product states.

T is the total task which may include several sub-tasks (t_i) and sub-sub-tasks (t_{ij}), for example:

- The first sub-task t_1: increment of the wheel number: $4 \rightarrow 8$
- The second sub-task t_2: to meet a higher standard of environment protection: Standard 1.0 → Standard 2.0
- The sub-sub-tasks of t_2 are:
  - The first sub-sub-task t_{21}: utilization of another type of power mode: petrol power → hybrid power of petrol and electricity
  - The second sub-sub-task t_{22}: realization of a better equipment for emissions
- Etc.

Consequently, the product development process can be described by a series of state changes. Given an initial state P_0, the product development process can be characterized by a sequence of product states « $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow \cdots \rightarrow P_n$ », where:

- P_1: The product state when t_1 is accomplished.
- P_2: The product state when t_2 is accomplished.
- P_3, P_4, etc.
- The final state P_n: all the sub-tasks are accomplished, in other words, the entire task T is accomplished.

Formally, task T can be characterized by a directed graph, defined as follow.

Definition 1. Task T is represented by a weighted directed graph $G(T) = (H, A, \Omega)$, where:

- H is a set of tasks, whose elements are the task T, the non-atom tasks t_m and the atom-tasks at_n, i.e., $H = \{h_i\} = \{T, t_1, t_2, \ldots, t_m, at_1, at_2, \ldots, at_n\}$;
- A is a set of directed arcs α_{pq}, i.e. h_p and h_q are linked by α_{pq}, from h_p to h_q;
- Ω is a set of weights ω_{pq} which are assigned to each arc α_{pq}.

In particular, the sub-tasks which do not have successors are named atom-tasks, noted as at_i.

A product development chain is illustrated by Fig. 1.

The task T is characterized by a graph, not a tree. In fact, there may be several sub-tasks which are not independent and they may have one or several sub-tasks in common. Characterization of knowledge K is based on the approach from Xu and Bernard [31]. The approach considers both the static features and dynamic changes of knowledge. For the static features of knowledge, a vector is used to help characterizing different aspects of knowledge such as quantity, granularity, compatibility and maturity. Such characterization mainly helps in dealing with explicit knowledge, for example, design knowledge organization, knowledge acquisition and storage. For the dynamic issues concerning knowledge evolution, the concept of knowledge state is applied. It describes the knowledge activities with state sequences. This is especially useful for processing product designers' knowledge, both explicit and tacit.

2.2. Knowledge value for product development

Supposing that knowledge K is necessary to accomplish the task T and a knowledge fragment k_i is needed to accomplish sub-task t_i, thus, k_i is the solution for the sub-task t_i, and knowledge K can be considered as a set of solutions which together can accomplish the task T. A knowledge fragment k_i can be a person, a book, a plan or any type of solutions provided.

Given this model, some questions may be: What knowledge K can accomplish the task T completely? If knowledge K can only solve a part of the task T, which part is solved? What knowledge fragments k_i have to be added in order to solve the remaining parts? How to choose the knowledge fragments k_i to accomplish the unsolved sub-tasks?

In order to answer these questions, some hypotheses are presented:

Hypothesis 1. The atom-tasks are noted as at_i, and all atom-tasks correspond to an explicit answer "yes" or "no" which shows whether it can be solved or not. In other words, the atom-tasks cannot be solved partially.

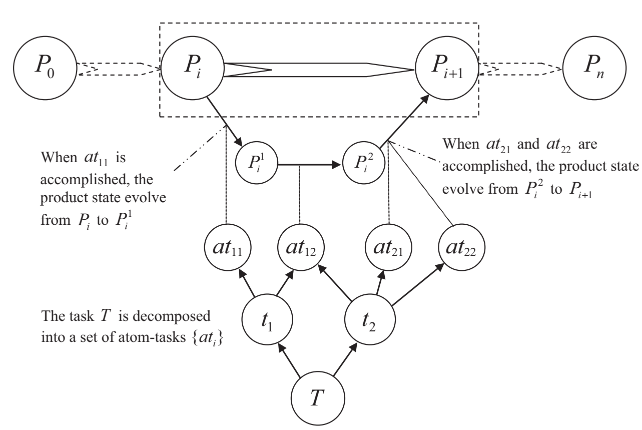

Fig. 1. A product development chain.