

A Study of Version Control for Collaborative CAD

Zhiyong Chang, Jie Zhao, and Rong Mo

The Key Laboratory of Contemporary Design and Integrated Manufacturing Technology,
Ministry of Education, China at Northwestern Polytechnical University, Xi'an
changzy@nwpu.edu.cn

Abstract. Version control is the prerequisite of collaborative product development environment, which, in its turn, has attracted increasing attention of CAD/CAM engineers. We aim to present our research results on a strategy of supporting evolution of versions in a collaborative environment. The strategy we provide is consists of a functional model which make application developing more easier, a deployment model in which a Version Management Unit is the deploying cell and a communication agent is included to exchange version information, a storage model which support that version information is saved in space of personal, group or global respectively. The topics discussed also in this paper are definition of version status, the transform of version status, the storage location of version and corresponding version operations.

Keywords: collaboration, version control, model.

1 Introduction

In a general way, the development process of product is consisting of a serial of sub-process in which some local result, such as drawing, 3D model, CAE analyze, is formed. Because of the non-linearity attribution of those sub-processes in whole process of product development, so development process is a complicated one which can be identified by attempt and reiteration[1]. The results mentioned above, which express a certain state belonging to whole lifecycle of product development, can be formalized and managed by means of version.

There are researches which show that engineer spends lots of time to search relative information in which version information is an important part. Although configure management and version management are the important and necessary function of PDM system, but the most of commercial PDM has only a simple version module which is very difficult of supporting development process of complicated product which is familiar in fields of aero-engine, airplane and in which engineer is trying, modifying, negating and re-trying. It is hard to image that engineer can work effectively without the support of a nice method and tool of version management.

There are more complicated problems involved in collaborative product development than in traditional product development. So in this paper, we present architecture of version management which aims at supporting collaborative product development. The remainder of the paper is organized as follows. In section 2 we briefly review the related works. In section 3 we detail the architecture from three aspects. In section 4 version operations is given. Section 5 is conclusion.

2 Related Works

The successful applying of CAD system makes a huge industry from 1980, then researchers realize that CAD system has non-regular requirement for data storage and regular DBMS(such as relational DBMS) is not suit for managing CAD model any more. So version management is presented as a available solution for the requirement of storage of historical development data, description of multi-view and multi-domain of product , hiberarchy of product data and so on, and it become a important research field recently.

Haskin and Lorie[4] from IBM present an extended relative model which has the ability of complicated object combination and supporting to long-term transaction to save and manage engineering data. Although their research is not associated directly with version management, but many later research of version is based on their works because of their large contribution for presenting combined object in relational DBMS.

McLeod[5] develop a semantic based data model framework, which use a serial of relative table to express semantic model, to support version management. Landis[6] focuses on version management for such complicated system as CAD by means of ontology method. The method based on four conceptions: non-linear version history, version reference, change spread and restriction of spread. Chou and Kim[7] give a new model extended from model of Batory and Kim[8], the new model emphasize the notifying of change spread. They present a message based notifying policy and a sign based notifying policy in which message about change is sent either immediately or later and product designer can be notified the change which might affect their own model only when they re-inquire for the design object. Vines[9] presents a version model for change control in which four conceptions are engaged to control change spread. Firstly, Timestamp instead of version number is used as ID of version to connect event with succedent version clearly. Secondly, relationships among objects, which can be marked as change-sensitive or change-insensitive, are created distinctly to define the spread range of change. Thirdly, such change management objects as change query object, change notify objects are used to manage change. Finally, the changes are grouped by means of configure object.

Wang[9][10] discuss and realize a system which can manage version of engine database on a collaborative environment. His method makes a structure by using version dictionary, which include system dictionary and designer dictionary, to organize and save version. The operations to version include setting up current version, recovering version, merging version and deleting version. Zhang[11] gives a version management model which is based on such conceptions as card file and file serial. Also a dictionary is used to manage structure and restriction of file card. The storage of model is based on a relative DBMS.

Although there are large numbers researches about version management which aims at constructing a frameworks and give corresponding arithmetic of version operation, but the most of works is based on or aim at version control about software development. The version management aimed at managing CAD data on a collaborative environment of product development is less researched and reported.