CHAPTER 8

DATA MANAGEMENT SYSTEM FOR
DEVELOPMENT OF HEADLAMPS

8.1 INTRODUCTION

This chapter tries to formulate a framework for managing data while a headlamp is developed for passenger cars as per the requirements of OEM by a supplier. This framework is very useful in managing changes as needed by the OEM on the style. These can be understood by the supplier through the data model proposed and reuse of previous data becomes an advantage in addition to collaboration.

The system supports information exchange among the co-designers, forming a horizontal integration and information exchange between the manufacturers representing the vertical integration.

8.2 DESIGN OF DATA MANAGEMENT SYSTEM

In this work, authors have adopted two levels of flexibility (a) at the product model level and (b) at data management level. Figure 8.1 shows the mapping of the product model and data management systems against the various levels of product abstractions. Product modelling system is specific to headlamp design process while the data management system is a generic one. Integration of these two systems can be achieved by database integration and or user interface integration. Customisation of product data management
system involves creation of information specific to headlamps. It depends on the type of integration between CAD and PDM systems. Next section discusses the integration approach for CAD and PDM systems.

![Diagram: Mapping of product abstraction levels onto system strategies]

**Figure 8.1  Mapping of product abstraction levels onto system strategies**

### 8.3 CAD – PDM INTEGRATION

The data integration can be of any one of the various levels as (oh et al 2001) given below:

a) Manually re-input the data  
b) Data exchange using file  
c) Stand-alone databases but the data is updated automatically.  
d) Shared database
In this work, data exchange using file method is chosen as it is simple and easy to implement. As already shown in figure 4.2, the product modelling involves four stages, namely, customer requirement modelling, function – form mapping, multiple view modelling and CAD modelling. The customer requirement page is accessed through internet and is one of the user pages included in the PDM system.

Product modelling system is designed to address requirements of OEMs on shape of exterior surface of headlamp, type of drive (left or right hand drive), standard of illumination, etc. OEM’s order is identified with order number, against which the requirements and documents (CAD file) are stored in database.

8.4 PRODUCT DATA MANAGEMENT SYSTEM

The PDM system developed in this work is a tailored system for the specific use to handle product information to supplier in the development of headlamps. It supports the following functionalities which are critical to headlamp supplier. They are

i. store parts and assembly using the part / assembly numbers generated in the design stage

ii. document management, where, document Ids are system generated

iii. product structure management, where parts are documents are mapped into headlamp assembly

iv. collaborative product development

v. access control (the users can access based on their role) and

vi. visualization of CAD files
8.4.1 Functional requirements

The data management system is modelled using UML concepts where Rational Rose® enterprise edition is used. UML is a standard language for visualizing, specifying, constructing and documenting the artefacts of a software-intensive system (Booch, et al 2004). The PDM requirements are implemented using Rational Rose® software. Use case diagram is constructed for the requirements and interaction of entities within the system. Basic element in a use case diagram is actor and use cases. Use case describes a set of sequences, in which each sequence represents the interaction of the things outside the system (actors) with the system itself. The functional requirements of the system are shown in use case model given in figure 8.2.

![Use case diagram showing the activities against various roles](image)

Figure 8.2 Use case diagram showing the activities against various roles
Various objects playing in these use cases by the different users were identified. The use case diagram partially provides the system workflow and description of workflow. The PDM system requirements have been established by constructing classes for the information system. The relationship among the classes is modelled as a class diagram using Rational Rose®. The class diagram for product structure is shown in Figure 8.3. The

![Class diagram for product structure](image)

**Figure 8.3 Class diagram for product structure**

source code (template) obtained for part entity bean is shown in Appendix. This code is needed to develop applications further.
8.4.2 Architectural Requirements

PDM system essentially becomes tiered architecture similar to the architecture of world wide web (WWW). Typically PDM system involves three tiers namely, presentation tier, to allow users to access the system through a web browser, business logic tier to handle PDM functionalities and data tier containing database and vault. Java 2 Enterprise Edition of Sun Microsystems (J2EE) and common object model (COM) / .NET of (Microsoft) offer refined component model that find application in complex realm of data management. A software component is the smallest deployable part of a program with contractually specified interfaces. Interfaces are the means by which components connect. The methods specified in interface are a set of named operations that can be invoked by a client. Within J2EE, there are elementary components called enterprise java beans (EJB) which make the development of distributed, transactional and portable applications possible. Middle tier of PDM system consists of a number of EJB components and includes connection mechanism with the data tier. The presentation tier is implemented as java server page (JSP) or servlet components which enable the users to access product data via hyper text transfer protocol (HTTP).

8.4.3 Building business tier components for PDM

For deriving the logical component diagram, use case diagram is restructured by generalizing total activities. The product, part, document and user are objects in the PDM systems. The user object interacts with other objects with the support of system infrastructure. The use cases with which the user interacts are user interaction layer. These use cases are important for design of user interfaces. Figure 8.4 shows the generalized use case view arranged logically.
Generally, component technology makes use of classes to capture default initial state and other component sources. A component has no externally observable state as in the case of classes. They can also be deployed independently. Since the architecture of PDM system uses J2EE, components are refined to such a level that loosely coupled EJBs are created.

Development of components is important in the design of component based PDM systems. Classes in product structure class diagram can be converted into EJB components by constructing interfaces depending on the service to the user. The application client can access only the methods declared in the remote interface, while access of web components and the
access of other EJB components are possible via both remote and local interfaces. For constructing components Rational Rose® was used.

8.4.4 Building data tier for PDM system

Data tier, at the back end of the architecture, is used to store product data. In this work, a relational database providing a complete functionality for data storage and retrieval is created. Oracle 9i® is used as database management server. Database connectivity provided by J2EE architecture, known as java database connectivity (JDBC) is used to access the database. The entity beans are linked to the relational data tables in the database. The database also supports file system for storing files.

Ideally, a PDM system is more about databases and data exchange. Product data repositories are operated and maintained by the user who is responsible for the data at that point in the product lifecycle. In product development process, there are disparate and heterogeneous systems that use same information partly or completely. The system integration strategy can be point to point translation, using shared database, using product data exchange standards or using application program interfaces (APIs) (Oh et al 2001). A typical engineering database should contain information that allows users to browse the database to understand the relationship among parts and products easily, such as bill of materials, part ID, part name, part drawings, etc.

The EJB components of the PDM system are the deployable elements of the software. In this work, Netbeans 6.1 Java editor for editing the codes is used. EJB was built in three tier architecture. For example, all the components of headlamps are different parts. The part EJB is created from part class, the part bean is accessed by clients via interfaces which may be a remote or local types. Figure 8.5 shows part class, part EJB and part object
definition page as shown below. The structure of the source code used to develop the part EJB is shown in Appendix.

![Diagram of Part and PartEJB]

a) Part class b) Part EJB created from the ‘part’ class c) Web based user interface for creating part

**Figure 8.5 Development of EJB component**

### 8.4.5 Essentials of the proposed PDM system

The data management system is designed for roles to which the users are mapped. The various roles are: administrator, designer, manufacturer, customer, supplier, detailer, reviewer and assembler. Administrator’s role organizes all the activities including allotment of user names and passwords for users. PDM system provides various functionalities for managing product data by the user. Table 8.1 shows the various roles, objects and activities involved in these functional modules.
<table>
<thead>
<tr>
<th>User (Role)</th>
<th>Activity</th>
<th>Object</th>
<th>PDM function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>Create, Delete, Modify</td>
<td>User Profile</td>
<td>Repository Management</td>
</tr>
<tr>
<td></td>
<td>Assign</td>
<td>Task</td>
<td>Workflow (Initiation)</td>
</tr>
<tr>
<td>Customer (OEM)</td>
<td>View status, Comment, add document and view parts</td>
<td>Part and Product objects</td>
<td>Supplier – OEM integration</td>
</tr>
<tr>
<td>Designer</td>
<td>Create, Modify, Delete and Update</td>
<td>Product, Part, Document, Version and search</td>
<td>Information storage and retrieval; Version management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Product-part mapping</td>
<td>Product structure management</td>
</tr>
<tr>
<td>Detailer</td>
<td>View parts, markup, check-in check-out</td>
<td>Parts and products objects</td>
<td>Collaborative design</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>View design, mark up models, counter proposal</td>
<td>Part objects</td>
<td>Collaborative product development</td>
</tr>
<tr>
<td>Assembler</td>
<td>View design, mark up models, counter proposal</td>
<td>Part objects</td>
<td>Collaborative product development</td>
</tr>
<tr>
<td>Supplier</td>
<td>View design, mark up models, counter proposal</td>
<td>Part object</td>
<td>Collaborative product development</td>
</tr>
<tr>
<td>Reviewer</td>
<td>View design, check-in, check-out, Approve</td>
<td>Part object</td>
<td>Release management</td>
</tr>
</tbody>
</table>
Designer's role allows the user to create, modify, and delete information on products, parts and documents. The system also permits visualization of the CAD models using STEP and VRML formats. Visualisation software allows markup. Designer uses the part number and product numbers which are automatically generated from the CAD package. The documents identification number is automatically generated by the PDM system.

OEMs can place order through the customer’s page. The OEM’s page allows specifying major requirements like the type of standards of illumination, type of drive, etc. OEMs can attach CAD files of car body profiles along with the order. The web pages for manufacturers, detailers, and reviewers allow the user to access files in their respective ‘inboxes’.

8.4.6 Visualisation

In today’s product development, there is a need to communicate design concept and share the design data interactively. Visualization module of this work is using virtual reality markup language (VRML). As mentioned early, geometric data are in STEP format for interoperability. Designers may choose any commercial software supporting STEP. The STEP files are converted to VRML. This format is light weight and can be embedded with standard HTML files.

For visualization, the client system need to install proper tools such as Autovue®, X3D and Java 3D. In this work, Autovue® of Cimmetry systems is used.
Figure 8.6  Visualization of front fender surface

Figure 8.6 shows the web based visualization showing car body profile and axis developed for the designer of reflector using Autovue®. This information is critical in design of headlamp.

8.6  SUMMARY

This chapter has discussed the development of a software system enabling the product development team to access the relevant data and create knowledge pertaining to his domain. Geometric data are exchanged in standard format ensuring interoperability and are stored in PDM system. Metadata and non-geometric data are also stored directly on the database. For visualization, the STEP file is converted into VRML format and is linked to HTML pages. Relational database management system is used with Oracle 9i software. The results and snapshots of the software developed is given in the next chapter.