Abstract - This paper describes the application of the co-operative design System (MecaCodemo) to optimize the design of the product from a multidisciplinary perspective. This system is developed at the Sols, Solides, Structures Laboratory at Grenoble and Polytechnic University of Bucharest. This application demonstrates an approach to solve many of the problems of interoperability, heterogeneity of platforms, and data sharing. Network connections utilize the ILOG software and are based on the CORBA standard. This environment enables the achievement of concurrent engineering goals. It gives the members of an enterprise-wide product development team a convenient, uniform interface to the global product data from different platforms. The different tools may also reside on heterogeneous platforms geographically dispersed.

Keywords – co-operative design System, ILOG software, CORBA standard

1. State of the art

Common to competitive enterprises today, is the quest for designing products and processes to meet a large variety of customers needs, in short time, based on few resources. Many of these products are unique, some are derivatives of existing products, while others are integration of a number of different products. At the same time, the market demands more and more customized products, resulting in an increasing number of products and variants. A critical aspect in the creation of these products is the understanding of an external user perception [1]. Quantification of these understanding servers as the decision base for the quality properties to be synthesized in the product [2]. The Design is the creation of an understanding of a product based on our knowledge [3]. The model shows how the real world, the model and the decision making are connected. The model is a projection of the real world, the data collected in the model forms the basics for the decisions making, and the decisions will, when carried out, give and effect in the real world.

The integration issue has been focussed for numerous years by 4 principal approaches [12], [13]:
♦ Design for eXchange,
♦ Value Analysis
♦ Integrated System of Design
♦ Co-operative Design systems

Design for X: Design for Manufacturing and Design for Assembly where introduced by Boothroyd and Dewhurst a long time ago. Many other Design for X (X=quality, cost, reliability,.), where X represents each professional trade which has to intervene during the design activities, have been developed until today and are largely used in industrial product development. The principle is basically the optimization of two professional trades. Considering the whole production system, Design for X addresses only a local issue of the cooperation of two engineers. The limits of Design for X are numerous. It gives greater place to a particular point of view (manufacturing, assembly, cost, reliability, quality) to the detriment of the other participant of the design process; all the points of view have to be considered in an optimal design.

Value Analysis and Quality: For forty years, those methods have been exploring and improving. The principle is to satisfy the customer who has the greater place to the detriment of the manufacturing processes; those methods are external-enterprise based. The integration of the different participants of the design project is realized through a work in interdisciplinary groups in which each participant brings his/her knowledge and constraints to build tables and criteria to evaluate the design solution. The limits of those methods come from the exclusive requirements of the customer in the definition of the production system. The internal aspects of the enterprise are disadvantaged in front of the external aspects due to the customer. Those two aspects have to be well balanced for an optimal system of production to satisfy both the customer and the enterprise.

Integrated System Design: Advanced techniques permitted to develop numerous applications able to
help design and control of production systems. Integration is here seen as a workflow of data from one application upstream to an application downstream. Data are integrated in databases joint and available to all the design participants. Applications specific to professional trades are also integrated into a design process considering a sequence imposed by the availability of input data. The limits of this approach are due to the prominence of computer implementation opposed to the flexibility of the design process itself. The design process able to chain all the applications is rigid, planed and pre-determined for all the problems of production system design and control. It does not permit to perfectly be adapted to each design process which is obligatory different from the previous one and to innovate easily.

Co-operative Design system: Today’s and tomorrow’s product development is integrated with different disciplines at a high degree, i.e. marketing, product design, manufacturing system design, manufacturing, sales, service and after market. The key is co-operative sytems, a tool for the extended enterprise based on an open information model [6]. Partners in the extended enterprise need to exchange legacy data and migrate with other systems outside their own secure corporate boundary [2], [3]. In order to achieve collaboration between different actors in the Extended Enterprise, there needs to be common processes supporting the distributed product development process [10].

2. Mechatronic Co-operative designModeller (MECA-CODEMO)

Companies work in collaborative processes within extended organisation to a large extent which means that information concerning product definition needs to be available, communicated and transferred in a not homogeneous information system environment. Today’s and tomorrow’s product development is integrated with different disciplines at a high degree, i.e. marketing, product design, manufacturing system design, manufacturing, sales, service and after market [2]. There is a large number of commercial software to support most of the activities within the product realization process.

To support multidisciplinary optimization a base definition must be defined as a common ground among the product development actors [2]. The base definition contains two major types of information, an entity hierarchy and an entity attributes. The ability to work with diverse multidiscipline teams members to successfully reach a goal objective. For a multidisciplinary group, the language of the system must do is independent of the languages of the disciplines, but is common to all. Thus, functional language should be a language of choice for multidisciplinary communication. Based on the integration of entirely separated department within the development a mechatronic product it is possible to define an enhanced process for the collaborative development of product properties in partnerships.

We propose a framework for interdisciplinary communication in the conceptual design stage with multidisciplinary optimization [2]. In order to facilitate effective communication across disciplines, this system argues that an integrated software framework should enable sharing and capture of multi-criteria design proposals, design semantics, critique, explanation and change notifications.

This System can be developed in two models alternative:

a) In the first model alternative, the different team members will perform varying engineering analyses with local compute resources. These local compute resources include hardware such as workstations and PCs and software such as simulation and analysis tools. Different team members need access to the some of the same computational analysis servers such as an element solver on a supercomputer, besides the process management and server utilities. Their tools should be integrated with the tools of other team members and the shared servers through the common intelligent repository of product and process data and agreed-upon protocols.

b) The second model alternative of product development environment considered was a fully distributed development environment (There would not be segregated local computing resources for each team member). All computing resources would be distributed across the network (Through an entry point to the network all simulation and analysis tools may potentially be accessed).

The product data to be browsed consists first of all of a base definition and engineering views (see figure 1 and 2).

To support multidisciplinary simulations, a base definition must be defined as a common ground among the product development team members [3], [5].

![Fig. 1 The engineering views](image-url)
The base definition contains (see figure 2) two major types of information (an entity hierarchy and entity attributes):

- The entity hierarchy describes how the components of the system are grouped together.
- The entity attributes for a part include, for example, moments of inertia, material properties, etc.

The design parameters are considered as attributes of entities in the base definition, and remain with the entities when they are regrouped in engineering views to create assemblies.

The feature-based design parameters serve as a common language to support design trade-off across various engineering disciplines where relevant performance of the mechatronic systems is measured.

As support of the design activity, a Co-operative Design Modeller [8] uses a client-server structures based on Ilog [6] product libraries with C++ object classes. The product model is composed of a knowledge model (features and production rules) and of data model (components, links and relations).

The server process has a usual function of storing data and distributing information, but it is able to do this among different operating systems running on different platforms. We have as much client processes as we have participants and each of them allows the database access, the multi-views and multi-representations of data and the connection to specific trades applications. Coupled to Remote Procedure Control (RPC) connection, IlogServer libraries are used to propagate all actions from server to client or from client to server. The features translation is one of the automatic tasks managed by the internal actor. The product model is progressively build associating the knowledge model (the feature) to the data model (components, links and relations). The different actors can use a set of predefined features, some of them being vernacular and for their own purpose some others being vehicular or universal and so having to be shared. Some rights are given to specific actors in order to read, to add or modify a vernacular or universal feature. A main interest in using feature is to be able to declare the instance of this feature without to be immediately obliged to define the values of their characteristics. Instead of specifying a value for a characteristic, an actor can use the constraints on this characteristic. An other mechanism, which allows constraining a characteristic, is in the setting of a relation on this characteristic. This is possible if the characteristic has been associated with a link of the component that is concerned. Design parameters are associated with the dimensions of features in the parameterized computer aided engineering (CAE) models. The design parameters are considered as attributes of features in the base definition and remain with the features when they are regrouped in engineering views to create assemblies [9].

Due to the fact that data requirements vary from discipline to discipline it must allow the engineers to augment the data model in the base definition with discipline specific data. While allowing diverse data to added it has also to maintain the consistency among these data so that the common ground is not broken and design trade-off across different disciplines can be performed. A key concept to address these issues, the engineering views, is introduced in the infrastructures of system. The major functionality of the engineering views are view model creation, data consistency with base definitions to support multidisciplinary design trade-off and mapping definition to automate view model creation during design iterations. As the translation task propagation and coherence management have to assist each actor. Part of view model is the construction of an entity hierarchy starting with the database definition entity hierarchy according to the needs of the particular engineering discipline at the actors (figure 3). The actor executes the simulations and analyses in his working environment. The “product model” is composed of a knowledge model (features and production rules) and of data model (components, links and relations). The different actors can use a set of predefined features and those can be shared. Some rights are given to specific actors in order to read, to add or modify a particular or universal feature.
A main interest in using feature is to be able to declare this feature without to be obliged immediately define the values of their characteristic. Some rights are given to specific actors in order to read, to add or modify the feature. A main interest in using feature is to be able to declare the instance of this feature without to be immediately obliged to define the values of their characteristics. Instead of specifying a value for a characteristic, an actor can use the constraints on this characteristic.

3. Conclusions

The MECA-CoDeMo system implements the product data model and stores the product data in heterogeneous, distributed databases. The product data model has been designed to capture the engineering data needs of the individual workspaces. The system enables the different team members to retrieve a wide collection of mechatronical system data to start engineering analyses, and store the analysis results. Other team members, thus allowing for a collaborative design of the mechatronic system can use simulation results stored by one team member. The system provides a catalog of independent objects, i.e., models, simulation scenarios, analysis results, and the product configuration hierarchy, stored in the global database and managed versions of each of these objects produced during the course of the design analysis and development process. This architecture demonstrates an approach to solve many of the problems of interoperability, heterogeneity of platforms, and data sharing. Network connections utilize the ILOG [6] software and are based on the CORBA [7] (Common Object Request Broker Architecture) standard, distributed object architecture. CORBA allows processes to share objects even over a network. Using the object technology of Corba makes the application flexible and extendible. It allows their various simulation and modeling tools on the different platforms to inter-operate through the Co-operative Design System [8], [12], [13]. One of these goals is to bring engineers from the various disciplines, servicemen, and even customers early in the product development process to access design of the product concurrently [14], [15], [16], [17], [18]. As support of the design activity, a Co-operative Design Modeller [9] uses a client-server structures based on Ilog [6] product libraries with C++ object classes. The product model is composed of a knowledge model (features and production rules) and of data model (components, links and relations). An applet provides a convenient user interface capable of being launched from almost any platform with the proper network connections. By being individually interfaced with the MECA-CoDeMo, the various tools and workspaces can inter-operate even from heterogeneous platforms.

References

Manufacturing paradigm for the extended entreprise in the new digital economy

Pugh, Addison-Wesley Publishing Company, Reading, MA, USA.