Intelligent Agents for Smart Systems

Amol Deshpande[1] and Richard F. Riesenfeld[2]

[1]University of Utah, Salt Lake City, Graduated, Currently Working
[2]Professor, School of Computing, University of Utah, Salt Lake City, UT

Abstract

Intelligent Design Assistants (IDAs) are smart agents that help in the solution of many complex computer aided design (CAD) problems. IDAs are based on the idea of providing the knowledge to the CAD environment to reduce the complexity of the overall process. IDAs represent another interesting application of the agent technology within CAD – Product Data Management (PDM) – Product Lifecycle Management (PLM) domain. IDAs are classified depending on their application and use. This paper discusses the design and use of an IDAs for various tasks within the CAD–PDM–PLM arena. The objective is efficient performance by alleviating the cognitive load and technical demand on the designer. The Gear Design Agent (GDA), one of the demonstrated IDAs, is constructed to perform the design data calculations for gears by providing associated design knowledge to the user. In addition to that, the GDA serves as a real-time information source for dependencies and attribute definitions.

Keywords: Intelligent Design Agents, CAD application, PDM, PLM applications, Gears,

1. Introduction

As a result of wide scale adoption of digital design tools in last decade, the CAD environment is facing new complex challenges. The design activity is no longer restricted to just designing a part using a computer. It has reached a level of complexity where the user expects complete assistance from concept to post production support level. In today's fiercely competitive world of product development, the complete design activity has become quite extensive. The PDM components have added a new dimension to the scale of complexity of CAD systems. In addition to the CAD-PDM integration, present systems, which also have to address the Product Lifecycle Management issues, require greater simplicity and ease of use. New features have inexorably been added to essentially the same conceptual framework. Some of the major challenges are concerned with managing the complexity of the overall design process and the life cycle patterns for the product being developed. Large numbers of parts, complex geometry, parts exhibiting multiple functions, ambiguous requirements, diverse knowledge and expertise associated with particular design are factors that add to the complexity of design. A methodology for addressing these issues is explained in the data complexity management concepts [3]. One of the ways to make it simpler is to make the system itself 'intelligent' and user friendly so that the requisite attention demands and skill level of the user can be reduced. Traditionally, a highly skilled user has always been an indispensable constituent of an advanced system. This particular need makes the overall product definition dependent on specific individuals –an undesirable vulnerability in the world of global cooperation and transactions.

There had been continual efforts to make design systems more efficient and user friendly, however, the need for the user's knowledge of mechanical and machine design could not be dispensed with. An Intelligent Design Assistant is one of the promising tools for a variety of design complexity problems that exist in a CAD-PLM environment. An agent or an assistant is a computer system situated in some environment that is capable of flexible autonomous action in order to meet its design objectives [1]. Intelligent Design Assistants (IDAs) manifest themselves in different types like *data agents, checking agents, monitoring agents, analysis agents, communication agents,* and combinations of two or more basic types. They are static as well as dynamic in nature depending on their functionality and requirements.

This paper presents a detail discussion on the role of IDAs in CAD–PDM–PLM systems. Two types of IDAs, namely, Design Data Agents (DDAs) and Interface Agents (IAs) are presented at greater detail here. DDAs and IAs work as integral parts of an experimental CAD system, Alpha_1.

Subject work presents a combination of the complexity management concepts [3] and the agent development activity [2] aimed at better system performance.

2. Need of Intelligent Agents

In any CAD application, creating the correct geometry of any part is a vital step, as is similarly the case for data associated with any design for a PDM – PLM system. The data is of various types, such as, part geometry, functional specifications, neighborhood information, dependency relationships, revision details, version details, and other related attributes. This variety of data is stored in different formats depending on the data type. Every PDM-PLM application enjoys the freedom of the way these different data types are stored and presented to the user. Generally, these different data types are identified as attributes in the CAD-PLM domain. The user needs to check for the attribute details to get specific information about any part under consideration. The unfortunate state of affairs is that, while customizing these applications, this variety of the data needs to be entered exclusively within the application interface. This activity is human dependent and prone to data entry error. In such cases, the user's knowledge is of importance in selecting certain key attributes.

Effective data management is the primary function of any PDM system. PLM systems use this data and offer a wider choice for product management. The key challenge is how to manage this enormous information space effectively. Since, the activities within PDM-PLM domain offer comprehensive solution to the users, smart handling of the information is necessary. IDAs are one of the promising approaches for such issues. The foremost advantage of the IDAs is that they posses the intelligence (in terms of data) within them.

3. Details of IDAs

IDAs are capable of playing variety of roles depending on the requirements. In order to address the issues mentioned above, a fleet of agents assigned to perform a designated task is needed. Each agent performs single or multiple tasks depending on its role and invocation context. The tasks include data generation, data collection, data monitoring, data entry, data representation, information sharing across systems, and the like, as needed.

The agents need to be imbued with the basic information to get started. Once the initial teaching/learning is done, then the agents work independently. This is somewhat akin to teaching a robot, particulars of a subsequently designated task initially. In case of a conflict situation the agents issue warning messages to the user so that an ensuing error/conflict can be avoided.

Presenting the details of every agent is beyond the scope of this paper hence they are mentioned only for reference. However, for better understanding, two complex situations where IDAs can be applied are discussed in further detail.

4. Design Data Agents (DDAs)

In any machine design problem, there is a basic need for the associated design data. Design data is calculated to be suitable for the purpose of the part under consideration. This involves input in terms of various parameters and working conditions. Depending on the requirements, it is the designer's job to generate data for the best possible design, one that is ready to manufacture. The appropriate parameter considerations come from the knowledge and experience of the designer.

The DDAs perform design data calculations and also provide associated design knowledge to the user in a highly interactive manner.

4.1. Gear Design Agent (GDA)

Gears are one of the most important and common mechanical parts in any mechanical assembly involving power transmission. Gears are meant for transmitting the power at the desired speed. Gear design is a very complex activity involving many input parameters and operating conditions. The GDA performs the design data calculations and also provides associated design information to the user in a highly interactive manner. The output from the agent is such that it can be readily passed on for generating a corresponding geometric model of the gear, which in turn can be viewed with TK3d graphical editor. The GDA is structured such that it can be used for a fresh design as well as in reverse engineering applications.

4.2. Design Considerations for GDA

Gears designed for any application are such that they will sustain acting loads at any given point of time and loading condition with a considerable margin of safety.

To ease the complex design process, a systematic approach based on the design principles and assumptions is followed. In GDA, *module* is the prime gear design parameter. A little consideration shows that module is the index of tooth size and ultimately gear strength, also. Another important feature is that most of the design parameters for a gear are normally expressed in terms of module.

In GDA, the basic functionality is to determine the minimum value of *module* using bending fatigue and surface fatigue stress calculations.

GDA is designed to work in the context where the working conditions and the specifications in terms of power or torque transmission and speed are known. The design algorithm used in GDA follows the guidelines of American Gear Manufacturers' Association (AGMA) standards with appropriate assumptions where necessary. The purpose of the GDA is to add knowledge to the application that traditionally would have been provided by a skilled user. Depending on the user's choice, the GDA selects the corresponding design algorithm automatically.

After running one complete cycle of the agent utility, the designer gets the output in the form of basic design parameters that lead to the generation of a geometry model. The utility presents the data in the format that can be readily used within the Alpha_1 experimental CAD/CAM system. The graphical representation of the designed gear can be viewed using TK3d. GDA can be used for spur and helical gear types. Figure 1 shows the screen shot from TK3d showing a spur gear model designed using GDA. Figure 2 shows the helical gear profile obtained using GDA from TK3d.

5. Interface Agents (IAs)

For any PDM-PLM application, interaction across applications is an imperative for effectiveness and ease of use. The interaction includes sharing data across different design environments. For every other application associated with PDM-PLM system, independent interface is required. These interfaces are already in existence and the PLM products have shown considerable flexibility in terms of using the digital design data across different CAD applications. In certain situations, the CAD data from one application needs to be converted to a type that is understood by other applications integrated with the PDM-PLM application. However, this kind of data conversion can be rather complex and may result in other compatibility issues. Collecting the attribute information for the CAD models created in different domains is a crucial, but too often a manual activity.

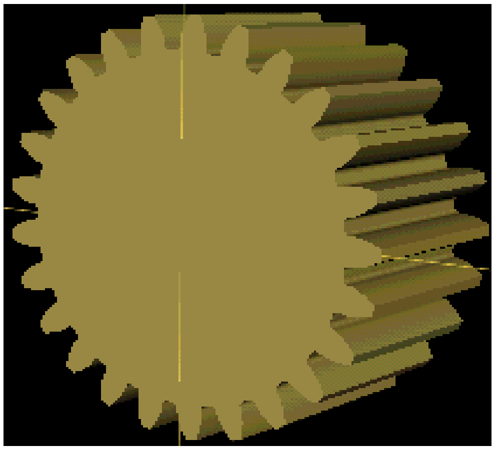

Figure 1: Screen shot for Spur Gear model designed using GDA

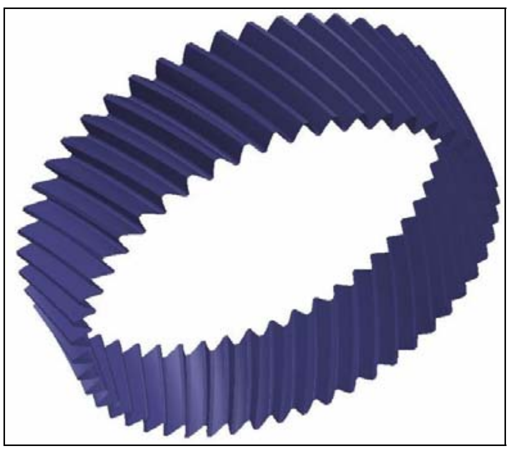

Figure 2: Screen shot for Helical Gear profile designed using GDA

The Interface Agents are second order agents. They are dependent on the first order agents (like GDA) for their functioning. The IAs utilize the information from the first order agents. However for the functioning of second order agents the system structure needs to acknowledge and allow the existence of agents.

5.1. Smart Design Document (SDD)

SDD is an agent that captures information from the identified CAD models that are created using the DDA(s) and populates the respective attribute fields. SDD is one place representation of these details for any mechanical part. Data stored in SDD and the data generated by the Design Agents is shared across to maintain a dynamic relationship with each other. In a mutual relationship between agents one agent serves as 'Master Agent'. The information maintained by the master agent is considered as the reference information in case of conflict. The user can choose which agent to designate as the 'Master' in a pair. For example the design data retained by SDD works as reference data for GDA for the corresponding gear design activity. In this case, the SDD acts as the master agent for the design constraints. The SDD holds information for the allowable tolerances or restrictions on the design in terms of mass, size, and location. The data generated by GDA is confirmed against the allowable limits specified in SDD at real-time

6. Features of IDAs

Apart from the automation of the design data calculation process, what makes Design Data Agents (DDAs) unique is the design knowledge offered to the user by means of an interactive GUI. Additional information about the input parameters is available at a mouse click. Related knowledge for any design parameter is made available to the user when asked for *More Info*. *More Info* section of the utility is the area where machine design knowledge is stored and gives an opportunity to reduce the need of skilled user as designer.

The DDAs have another advantage in terms of presenting information to the user that has certain advantages such as faster and better design with lower possibilities of design conflicts. The architecture of a DDA is very portable and can be altered for other design elements such as bearings, shafts, etc. The DDA is designed such that it can be used for a fresh design as well as in reverse engineering applications. Data generated by the DDAs can be used as source data by other agents (example Interface Agents) in the system. This can be looked as a multi-agent environment where the data generated by one agent is used for other agent activities. The information from agents is used to address the cases of design conflicts or dependencies. In such situations the data generated by the respective agents is accessed by the monitoring agent and the user is alerted to the possible conflict.

SDD is one place representation of the attribute details associated with given CAD model. One can choose to which level of agent to make as the master agent. The information fed into the master agent in the agent-agent relationship works as master information in case of conflict. After the one time teaching the agents are capable of repeating the same task without any error in the given setup. SDD is possible in different forms. Due to the loose integration property of agents, i.e. their modularity, it is relatively easy to integrate them into a system architecturally.

There are attempts to incorporate the knowledge base within the CAD system itself as in CATIA V5. They have introduced the features of Knowledge Based Engineering (KBE). However, DDAs are different from KBE though it may appear somewhat similar. KBE features are closely related to design refining tools like Dimension Measurement Tool, Ground Clearance Tool, Suspension Analysis Tool, Tire Clearance Tool, etc. [5]. These tools do not address the very process of part design thoughts. These tools are helpful for the full product details which may not otherwise be required for all the design activities. DDAs address the actual design process and provide the necessary design knowledge to the user. It does not work on top of already designed components to verify certain aspects of the product.

7. Conclusion and Future Work

IDAs are 'practical reasoning agents' that are inspired by a theory of pragmatic reasoning in humans used to decide what to do. DDAs can be viewed as a demonstrated example of IDAs in the CAD framework. Smart Design Documents is an example of second order agents and classified under Interface Agents. Second order agents are discussed in detail here for their functioning however, the actual implementation is at prototype stage due to certain issues with the different CAD domains available for the research. Similar agents can be created for other mechanical parts or assemblies that add the machine design knowledge into the software environment. These agents work as an integral part of the system but also have the capability to work in stand alone mode. This allows GUIs for DDAs to be inserted in web based applications also.

DDAs and IAs are classic examples of the automated intelligent agents as discussed by Jennings, Sycara, and Wooldridge [1] and W Shen, Jen-Paul, and Barthes [2], in the CAD-PDM-PLM domain. Autonomy in the design process is otherwise a difficult concept to pin down precisely, but we mean it simply in the sense that the system should be able to act without relying on the user skills and attention to perform and should have control over its own actions and internal state. We perceive it as a step in the direction of agent based CAD-PDM-PLM system structure conceptualized in terms of agents design and implemented by acknowledging the existence of agents. IDAs are more than mere programming functions that can be used across the system; it has an embedded, integral knowledge base. IDAs are capable of taking initiative in the complex design steps to make the overall process easier and liberate the user with the need of certain mandatory skills. DDAs and IAs fall in the category of 'expert assistants' operating as a semi-autonomous agents as defined by Jennings, Sycara and Wooldridge [1]

These different agents can be integrated together to establish the information network across the system. Most importantly these agents are easy and time saving to build than whole system addressing the similar needs.

Populating a fleet of DDAs and IAs within the existing system will make the system dynamic. Also, the network of such intelligent agents will be helpful in creating a web of dynamic Smart Design Documents. Other types of agents like analysis agents or monitoring agents e.g. can be developed along similar lines. However, for better results and efficient response, the system has to support the functioning of these agents within itself. For this purpose there is a need of building a system structure which includes agents in its core as native and integral elements. Such designed systems are expected to replace the existing systems and lead to the more intelligent environments we all seek.

Acknowledgement

This work was supported in part by ARO (DAAD19-01-1-0013). All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

8. References

[1] Nicholas R. Jennings, Katia Sycara, Micheal Wooldridge, 'A Roadmap of Agent Research and Development'.

[2] W Shen, Jen-Paul, A Barthes, 'An Experimental Multi-Agent Environment for Engineering Design'

[3] Jacobs, Tim, M. 1998, ‘An Automated Framework for managing Design Complexity Draft’ Ph. D. Thesis, University of Utah, Salt Lake City, UT.

[4] Baker, B. A., Fish, Russ D., Cohen Elaine, ‘Using Multiple Concurrent Design Views Interface to Enhance Design Complexity Management’, Proceedings ASME Design Engineering Technical Conference, ASME, DETC2000/DAC-14261, Baltimore, MD.

[5] CATIA V5, Knowledge Based Engineering (KBE) documentation

[6] Amol Deshpande, 2004, Smart Design Agents for Complex Systems with Integral Design Knowledge, Masters Thesis, University of Utah, Salt Lake City, UT

[7] Loss, B. F., 1997, ‘DRIVE: A system for Interactive Review of Engineering Design’, Masters Thesis, University of Utah, Salt Lake City, UT.

[8] Shigley, Joseph E. and Mischke Charles R. 2001, ‘Mechanical Engineering Design (in SI Units)’ sixth edition, Tata McGraw Hill Publishing Company Limited, New Delhi.

[9] Juvinall, Robert C. and Marshek, Kurt M. 1991, Fundamentals of Machine Component Design, John Wiley and Sons, New York

[10] Kapelevich Alexander L., and Kleiss, Roderick E., Gear Technology September / October 2002 ‘Direct Gear Design for Spur and Helical Involute Gears’, www.geartechnology.com .

[11] SHOPSWARF, New Zealand http://shopswarf.orcon.net.nz/spur.html

[12] S Rajagopalan, J.M. Pinilla, P. Losleben, Q. Tian, and S. K. Gupta “Integrated Design and Rapid Manufacturing Over the Internet.” presented at Proceedings of DETC’98, 1998.

[13] M.R. Danesh and Y. Jin, “ADN: An Agent Based Decision Network for Concurrent Design and Manufacturing,” presented at the proceedings of DETC’99, Las Vegas, NV, 1999.

[14] J. Shah, M. Mantyla, “Parametric and Feature Based CAD/CAM, 1995

[15] G. Pahl and W. Beitz, ‘Engineering Design – A systematic Approach:’ Springer-Verlag, 1998.