INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

AN ONTOLOGY BASED APPROACH TO A FLEXIBLE AID FOR MECHANICAL CONCEPTUAL DESIGN

Yang Liu and Anton H. Basson
Stellenbosch University, South Africa

ABSTRACT
Small and medium sized engineering companies involved in product development are under considerable pressure, due to globalization, to reduce product development cost and time. Since such small companies typically cannot afford customised design support tools, but employ a wide range of design styles, there is a need for "flexible" (i.e. easily adaptable) design support tools. Flexibility in this context indicates that the same software tool can be adapted to a wide spectrum of design styles without requiring changes to the software's source code. The approach presented here to achieve flexibility for distributed design support systems, uses the notion of an ontology, combined with elements and relations in conceptual graphs for the database and user interface. This paper presents the implementation of these aspects in a design support system called DiDeas II (Distributed Design Assistant). It is aimed at the pre-detail design phases, and aims to allow design teams to manage their design information according to various design methods, to decrease time-delays and to improve communication between team members. The main focus of the paper is the practical implementation of an ontology based design support system, DiDeas II, but the results of preliminary evaluation in laboratory case studies are also reported.

Keywords: Distributed design; ontology; concept design

1 INTRODUCTION
During design process, the designer has to collect, organise and filter information. Because of the variety of types of products, he/she would work with different kinds of information in order to accomplish his/her design work. Furthermore, due to today's competitive market where technology progresses at a very fast pace, product design is getting more and more complicated and requires that the development time should be as short as possible without losing product quality. The early design phase plays an important role in the design process and studies have shown that although the cost of design is usually less than 20% of the total cost in a product life cycle, up to 80% of the total cost is determined during the design stage [1]. To manage the design information in an organized way is therefore very important for producing a good design. It also decreases the time that the designer has to spend on searching for or exchanging information, and this time can be used more productively to develop creative ideas for the design of the product.

Different design methodologies are applied in different companies and the design process is mainly determined by the company resources, the team members' experience, the project type and the project scope. Basson *et al* [2] outlined the diversity in design processes in terms of scope, models, methods and procedures. There is no common design methodology to fulfil the different requirements from different companies, with the result that there are many design methods, such as those presented by Blanchard and Fabrycky [3], Pahl and Beitz [4], Ullman, [5], Ulrich and Eppinger [6] and Nigel [7].

A number of tools are available to support teams during the early design phases (such as [8], [9], [10], and [11]). Wang et al. [12] summarise concept design support tools developed in recent years. However, there is still an need for greater flexibility in these systems, providing the opportunity for teams to alternate between different design methodologies for different design resources and projects.

Ontology based approaches are widely applied where diverse information has to be handled. When combined with the notions of elements and relations from conceptual graphs, it has the potential for high flexibility. This paper presents an ontology-based approach implemented in an internet-based design support system, DiDeas II, which is a development of DiDeas I [13]. DiDeas II is intended for distributed teams doing mechanical design in the pre-detail design phases, and aims to allow design teams to manage their design information according to various design methods (according to their company's design style), to decrease time-delays and to improve communication between team members. The research is aimed at application in small to medium sized engineering companies where the cost of customised software for design support is prohibitive. An objective for DiDeas II was therefore that users should be able to customise it (e.g. by changing the design terminology or process) without having to change the software's source code.

2 COMMONALITY AND DIVERSITY IN DESIGN PROCESSES

2.1 Commonality in Design Procedures

Even though one of the objectives for DiDeas II is flexibility, it must still be identifiable as a design support system, as distinct from other information management systems. The elements that are present in virtually all design procedures should be built into DiDeas II. To determine these elements, a functional analysis of a general design process was conducted (Figure 1). The design process can be described as a map with instructions on how to get from the identification of a need for a specific object to the final product [5]. It can be said that this process transforms available information, knowledge and expertise in order to construct a means to get from an expressed need to a solution. An expression of a need as a starting point is clearly one of the elements found in all design procedures.

From the view of systems engineering [3], there is usually the need to decompose the target (system) into sub-targets (subsystems) to find solutions, with the result that decomposition, or subdivision of the design task, is a necessary element in a design support system. Baselines play the central role in this regard, in that a baseline is a common specification for the subsystems derived from it. The baseline must be such that if all the subsystems satisfy all the baseline's requirements (which can be expressed as functions and performance measures), then the system will perform the function(s) required of it. DiDeas II must therefore provide for subdivision of the design task and for managing the associated baselines.

Another aspect of subdivision, seen particularly in complex design tasks, is that the same design process is repeated on each hierarchical level of the decomposition, except at the last level where detail design and/or purchasing decisions have to be made. DiDeas II should therefore be applicable at any level of the hierarchy.

An aspect similar in some ways to subdivision is the generation of alternatives (design concepts) to choose from. From an information management point of view, concepts primarily differ from subsystems in that concepts represent alternatives, while all subsystems are required.

Another common notion in design procedures is iteration and a design support system must allow frequent changes to previously entered information.

The above notions are captured in the general design process depicted in Figure 1: The overall functions of the system being designed are the normal point of departure. From it, the overall objectives are formulated, followed by the generation of alternative concepts. Each of these alternative concepts can be decomposed into sub-concepts/subsystems, where the design process will be repeated (the dashed line block on the left hand side of the figure). The baseline ("Refine objective" in Figure 1) is pivotal in handling subdivision of systems and subsystems. The decomposition is continued until the sub-concept/subsystem level is reached where purchase/subcontract/detail design decisions can be made (the central, right hand side of the figure). Three typical design activities in sub-concept detail design are shown in Figure 1, i.e. component design, analysis using tools and layout design, but there can be many other design activities depending on the design project and design resources.

2.2 Ontology as a Strategy to Handle Diversity

In contrast to the common aspects addressed above, there are various aspects of design processes that differ from one team to the next or even from one project to the next. This section considers how to handle this diversity in DiDeas II.

One aspect of design diversity is the range of design terminologies, with the same word being used for different notions and different words used for similar notions [2]. This results in communication problems between designers that have different backgrounds, or confusion when using a design information system that employs terminology that the team is unfamiliar with. Each company typically has its own design terminology and design procedure, but small and medium sized enterprises cannot afford to have customised design support tools developed. DiDeas II is therefore aimed at allowing a company to easily adapt it to its own design style and ontology is the key to achieving this adaptability.

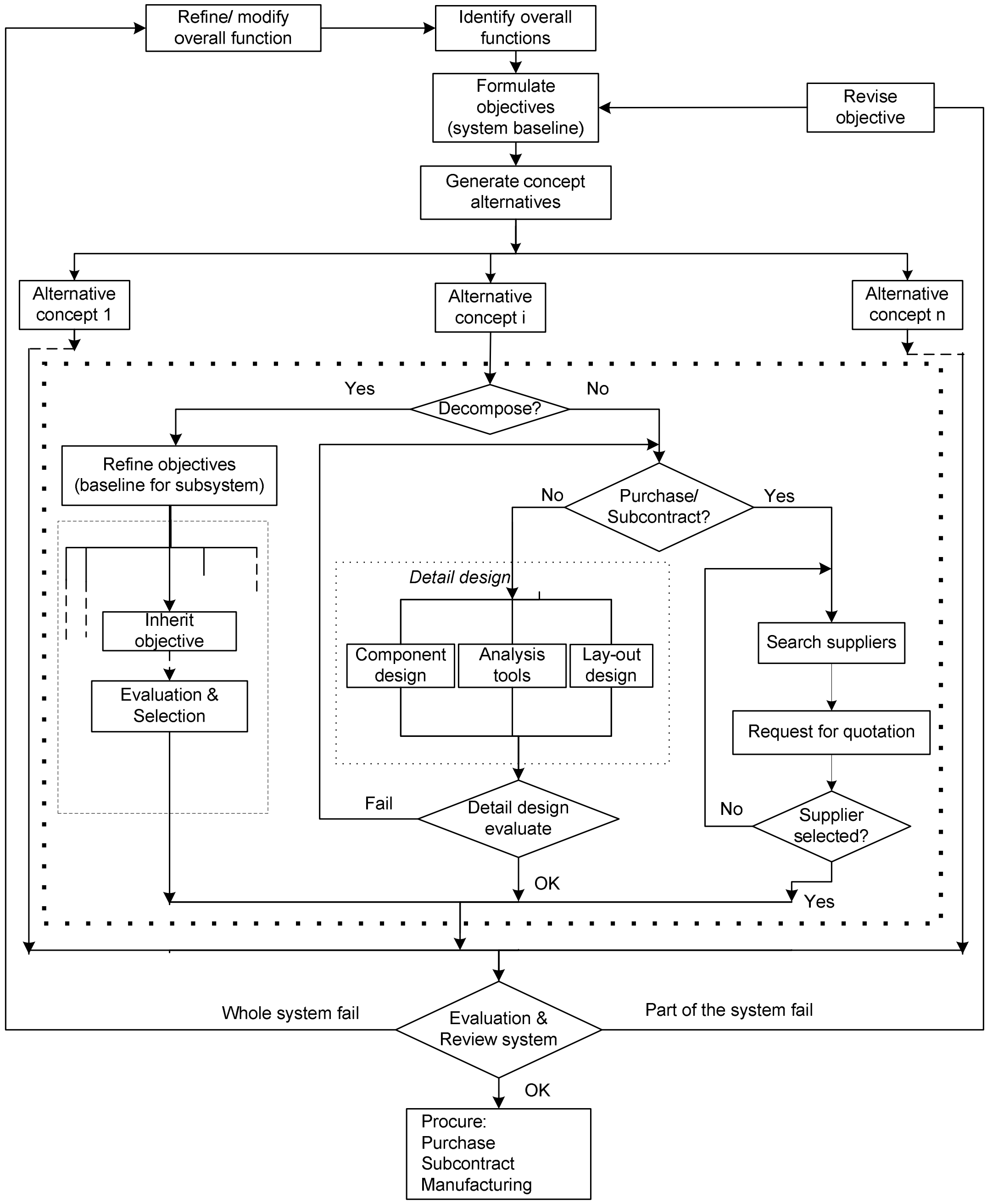

Figure 1. General Design Process

"Ontology" refers to the branch of philosophy dealing with the modelling of the reality. The use of ontology is becoming more and more popular in different research fields. There are many definitions of ontology [14], but the following general definition given by Neches *et al* [15] is useful in the present context: "An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules of combining terms and relations to define extensions to the vocabulary". The meaning of the ontology in this paper can be regarded as the conceptualization of design pattern according different design methodologies applied in different companies.

Conceptual graphs can be combined with an ontology-based approach to create a database structure. Sowa [16] gave the following definition of conceptual graph: "A conceptual graph is an abstract representation for logic with nodes called concepts and conceptual relations, linked together with arcs." In DiDeas II, all design information is classified either as an element or a relation. The elements

are only linked to relations, and vice versa (a property of a conceptual graph). The set of available types of elements and relationships constitute the ontology.
Two of the main databases in DiDeas II are (Figure 2) the Ontology Database, in which the element and relation types are stored, and the Project Database, in which the design project's information is held. Each piece of project information is classified as an element or relation of a type in the Ontology Database. Changes in the design process or terminology are implemented in the databases by adding to the Ontology Database, but the database structure does not change. This is the most significant difference between the approach implemented in DiDeas II and the typical relational database structure applied in DiDeas I, or even object orientated database approaches, which require prior knowledge of the design process to set up the database. In the latter cases, customising a design support tool typically requires changing the database structure, which significantly reduces flexibility.

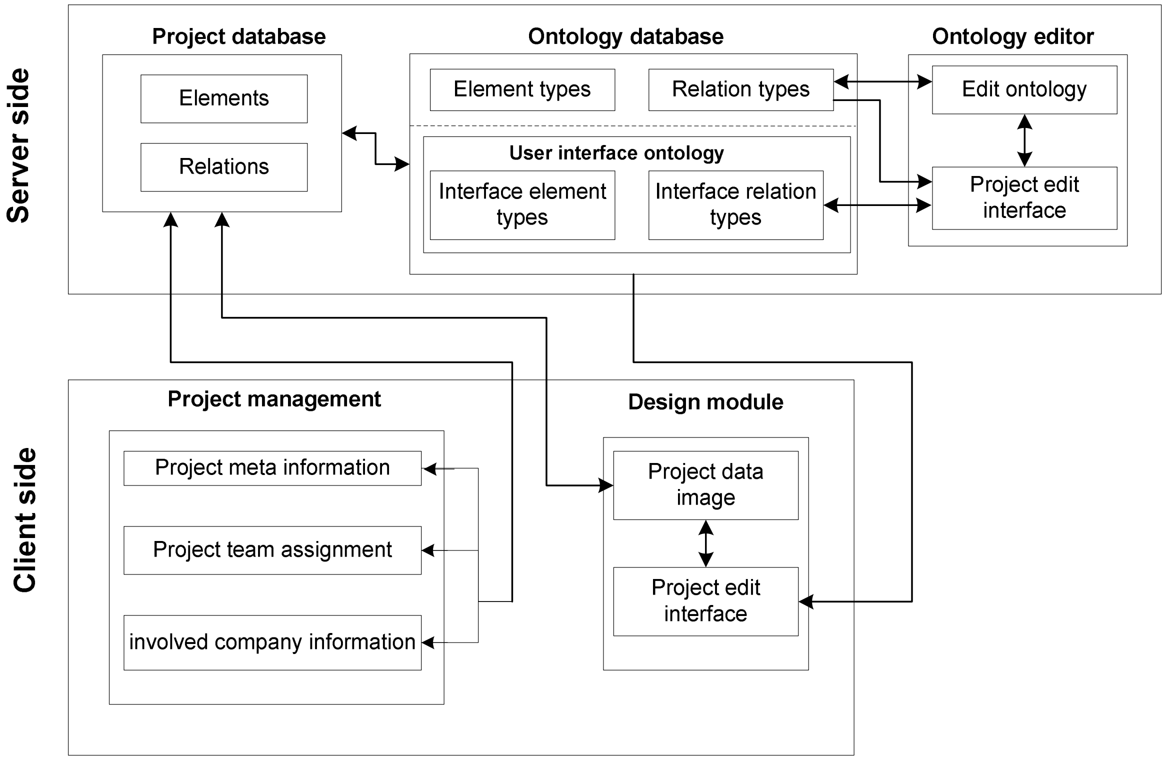

Figure 2: DiDeas II Architecture

3 DIDEAS I AND II

3.1 DiDeas I

DiDeas I [13][17] is an Internet-based system that allows simultaneous multi-user collaboration in the early design phases. In DiDeas I a relational database (implemented in Access) is located on a central web server and stores all design information entered into the system. The user interface was realised in the form of a collection of Microsoft Active Server Pages (ASP), which can be accessed platform-independently via a standard web browser.
Case studies showed that DiDeas I provides a suitable context for information exchange in synchronous and asynchronous collaboration scenarios. However, the use of HTML for the user interface severely restricted its flexibility and it is complex to adapt the relational database to a large variety of design styles.

3.2 DiDeas II

Several programming languages were considered for the development of DiDeas II, and Visual C++ with .NET was chosen since it imposed the fewest limitations. DiDeas II has two separate programs, i.e. one running on the server side and one on the client side (Figure 2). Each designer will run his/her own client side programme.

3.2.1 User Roles

As shown in Figure 2, there are different databases in DiDeas II. Three roles can be identified for the maintenance of the information in these databases, i.e. the normal design team member, the project manager and an "Ontology editor" who can be a project manager, systems engineer and/or chief designer. The ontology editor sets up the ontology database, thereby determining the design procedure and associated terminology that will be used. The ontology editor does this by maintaining the "Element types", "Relation types" and "User interface ontology" in the "Ontology database". Once this has been done, the normal design team members can start using the system to record and retrieve project information in the "Project database" through the "Design module" which reflects the design procedure and terminology set up by the "Ontology editor". The project manager maintains the overall project information and the team member allocation. This information is also kept in the "Project database".

The following sections describe the typical user interface screens through which the persons fulfilling the respective roles maintain the data in the various tables.

3.2.2 Server Side

The server side maintains the Ontology Database and the Project Database, as proposed by Basson *et al* [2]. The Ontology Editor module also runs on the server side and is used to create, edit and delete ontology elements and relations, as well as the Project Edit user interface. The client side cannot change the Ontology Database.

Figure 3 shows the user interface for editing the ontology element types and relation types. Each element type is assigned a unique ID, while the user can specify the element name and description. The relations types also each have a unique ID and description, but also a type classification (used in for the Project Edit user interface), the parent element type's ID and the child element type's ID.

Figure 4 shows the server side interface through which the user interface of the client side (which depends on the ontology element and relation types) is set up. Its use is described in the next section.

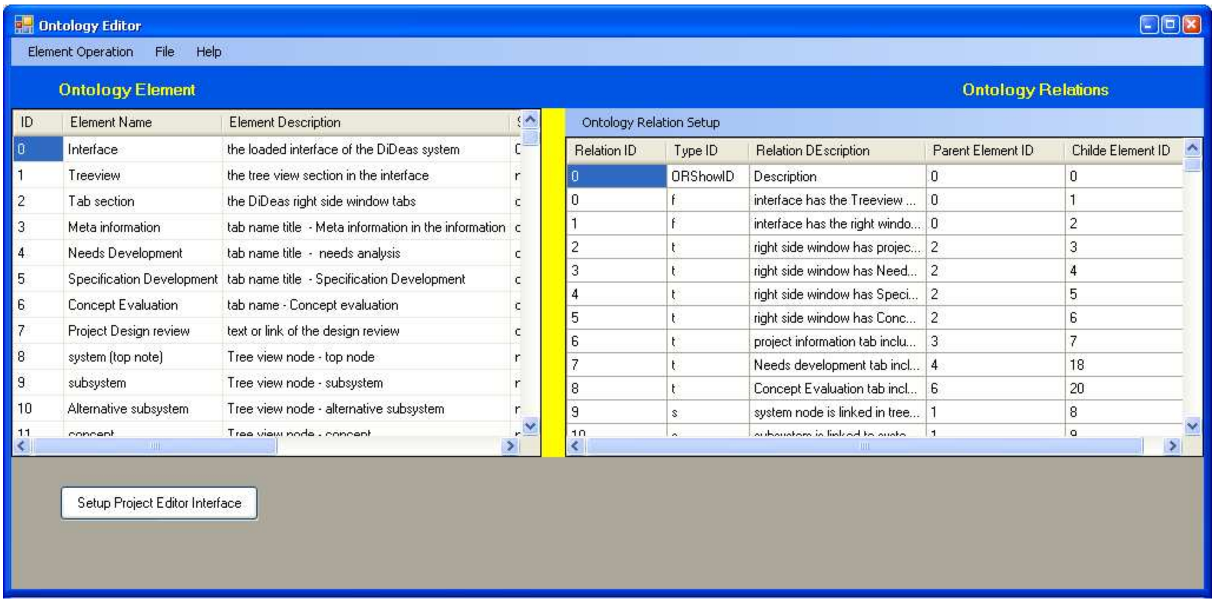

Figure 3: User Interface to Edit Ontology "Element Types" and "Relation Types"

3.2.3 Client Side Project Edit User Interface

The client side user interface allows the design team member to view and edit project information. It is customisable (an important element of flexibility) through the ontology defined on the server side. In the following discussion, the basic structure of the Project Edit user interface is first described and then the user interface on the server side used to customise it.

As mention above, an objective for DiDeas II was that the users can customise it (e.g. changing design terminology or process on the client side) without changing the software's source code. A balance had to be struck between giving the users the freedom to change the Project Edit user interface and keeping the process of changing the user interface simple enough so that people without programming experience could do it. After considering the various formats in which data can be displayed, the authors selected the following underlying user interface structure (illustrated in Figure 5):

On the left hand side of Figure 5, a tree view shows a break down of the system into subsystems and concepts. This information is common to virtually all design processes and was fixed in the source code. It is used as the means to navigate through the design information since all other information can be naturally related to one of the tree nodes. The top node always represents the project itself.

The decomposition is dominantly of a one-to-may character and the tree view is an intuitive way of presenting the information. However, in many projects a certain subsystem can be found in different parent subsystems. Provision is therefore made for a subsystem to appear more than once in the tree. The information that shared by more than one subsystems is highlighted in certain colour, and also a warning message is displayed when it is edited to remind the user that it occurs more than once in the tree. The detailed format of the display (colours, line types, etc) is customisable using the "Interface Style" tab shown near the top of Figure 4. Colours can also be used to distinguish concepts from subsystems.

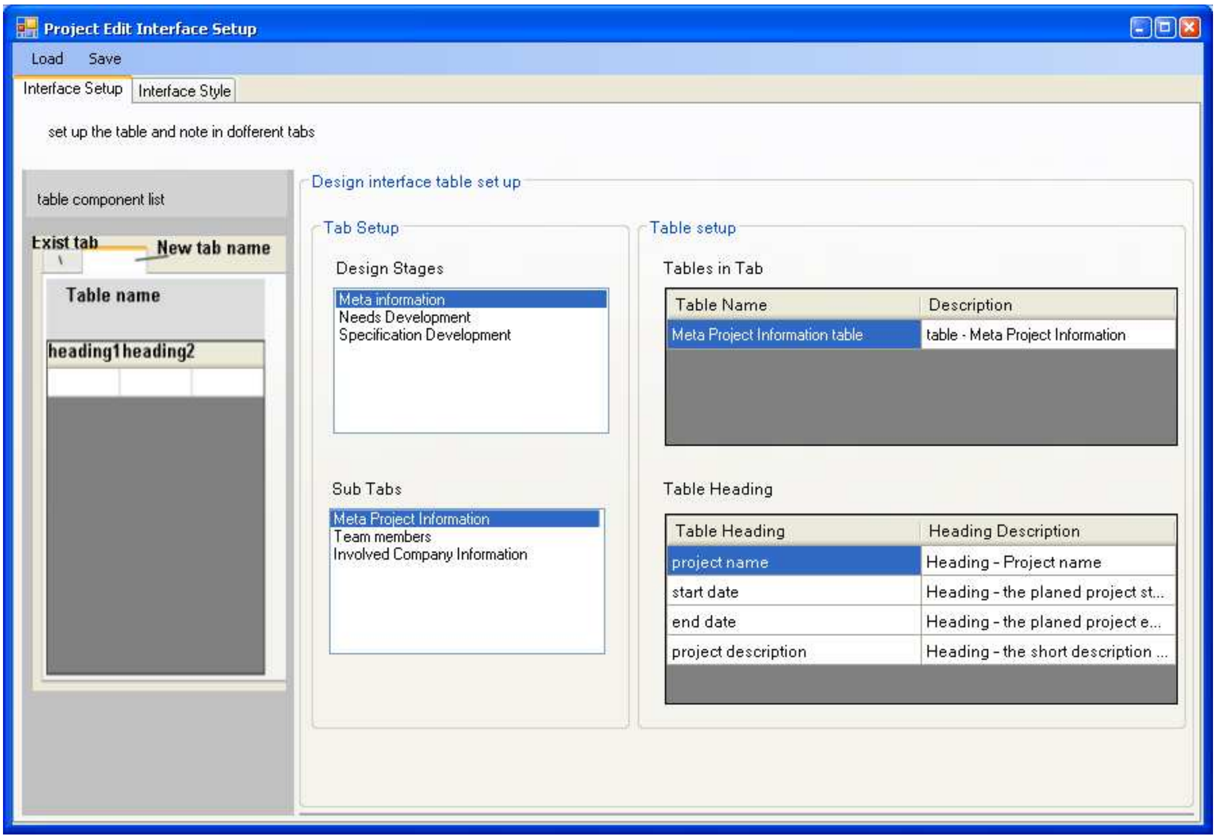

Figure 4: User Interface to Edit "User Interface Ontology"

The right hand side of the Project Edit user interface is built up out of nested tabbed pages containing tables that show information about the item selected in the tree (e.g. "Frame" in Figure 5), as well as its parent in the tree ("Support System" is the parent of "Frame" in Figure 5). The titles of the tabs and sub-tabs can be set using the "Tab Setup" boxes in Figure 4. The tabs are referred to as "Design stages" since they will normally represent the sequence of steps in the company's design procedure. Each sub-tab can contain one or more tables, which are set up using the right hand side of Figure 4. Each column in the table and the ontology element types that must be displayed in that column are also set in the "Table setup" section of Figure 4.

The information described above constitutes the "User interface ontology" in Figure 2. Figure 5 shows a typical setup for the Project Edit user interface, but all of the headings on the right-hand side and the information displayed can be customised through Figure 4. The tabbed pages give an implied sequence with two hierarchical levels, but without forcing the design to follow the sequence. The company's normal design procedure can hence be conveyed without forcing it upon the designer. The tables are suitable for displaying many-to-many relations, but all of the information showed in the tables on the right hand side of Figure 5 is related to the concept or subsystem selected in the tree on the left hand side (a one-to-many relationship).

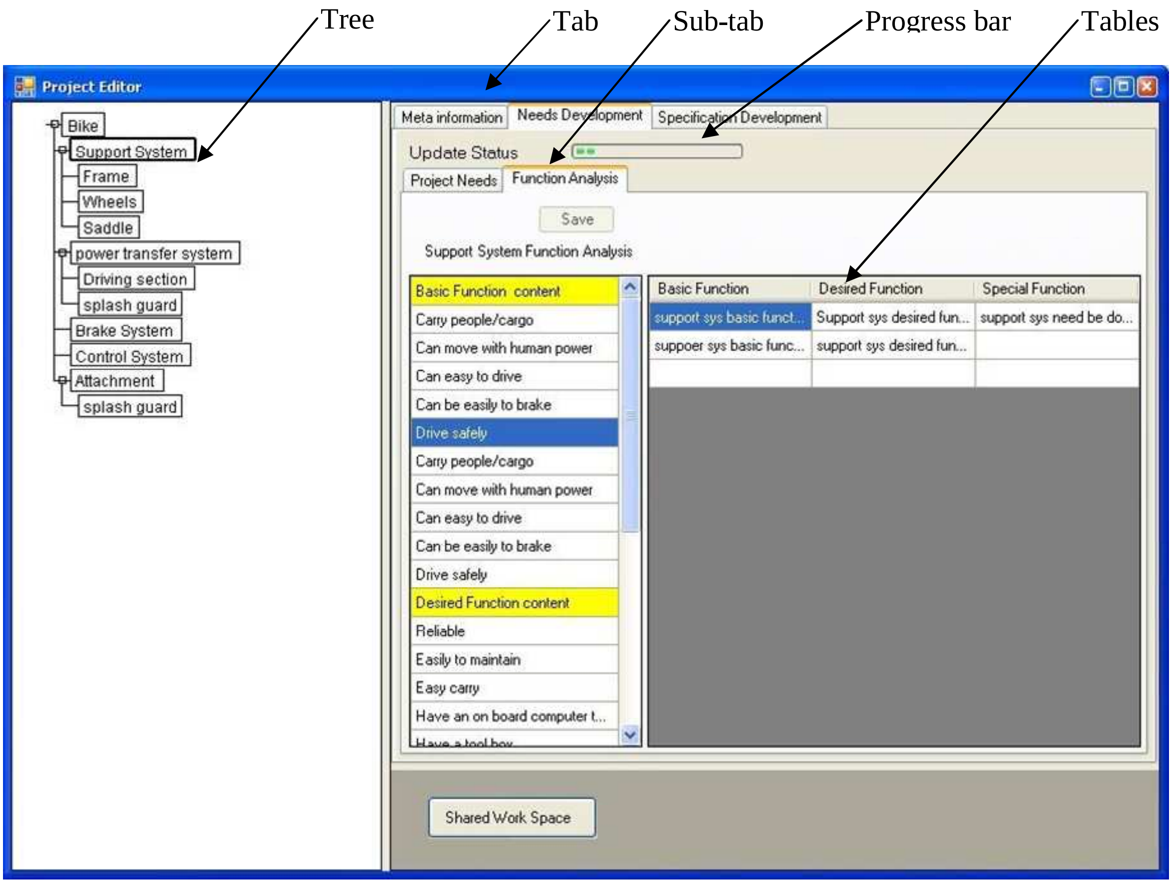

Figure 5: Client Side Project Edit User Interface

3.3.4 Other Client Side Aspects

When the client side software is first started, the user has to select a project. Only one project can be accessed at a time in the current version of DiDeas II.

Figure 6 summarises the data management process performed by the client side. The client side maintains a TCP/IP connection with the server side to exchange data with the server side Project Database. The client side maintains a local image of the project data to ensure rapid response of the user interface. Any information created or modified by the designer is saved in the image, but also immediately sent to the server. The server saves these changes in the server side database so that all the designers simultaneously working on the same project have access to the newest information. The client side programme periodically polls the server for updates generated by other designers. The "Update Status" progress bar in the Project Edit user interface (Figure 5) shows how much of the delay period before the next update, has passed.

Other user functions provided on the client side, which cannot be described in this paper due to length restrictions, are a QFD-style relationship window, a project management window (to manage project time scales, client company information and team member allocation), a shared workspace for uploading files to or retrieving files from the server, and a design review facility to manage design review records.

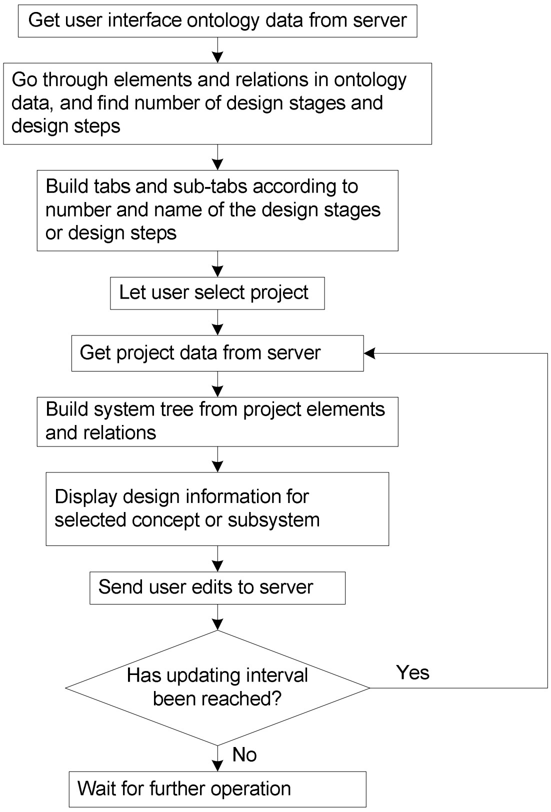

Figure 6: Flow Chart of the Data Management Process for Project Edit Interface

4 CASE STUDIES

Three preliminary case studies were conducted as a first evaluation of DiDeas II. Since these case studies were limited in scope and conducted in an academic environment, their results have a large margin of uncertainty, but they could still be used to give an indication of the success of the approach implemented in DiDeas II.

4.1 Case Studies One and Two

The first two cases studies were performed by students working in teams of three using an ontology created earlier by the first author. This ontology reflects the design procedure taught in the mechanical engineering programme. The students were from second year to masters-level in mechanical engineering that all had a similar education in design procedures, as well as two master's students from Process Engineering that had no mechanical design exposure. The teams were given projects such as the specification development for a coin sorting mechanism for a vending machine and the specification development of a bottle and can separator for a recycling plant. The students were all volunteers and spent a Saturday morning or afternoon working on the assigned projects. The team members worked in separate offices and could communicate only using DiDeas II and chat-software, except in the cases in the second case study where face-to-face teamwork was specifically employed. After the project they had to fill in questionnaires and informal interviews were conducted by the first author to elicit their responses.

In the first case study, the ability of DiDeas II to convey design information to persons that were not involved in creating that information, was evaluated. This scenario simulates where a new team member joins the design team halfway through the project, or where one team develops a design up to a point and another team takes it further or develops a follow-up project. It also indicates to what extent DiDeas II will aid communication between team members when the work asynchronously. The first part of this case study was for one team to set-up the specifications of a new project in DiDeas II, and to start with a subsystem breakdown. They were followed up in the second part by a different team that had to continue with the same project.

By reading through the information that was recorded in the Project Database, the second team could quickly get a clear view of the system structure in the tree view. Also the requirements and specifications entered by the preceding team acted as an example of what is required since they were linked to the different levels of the system-subsystem tree. Observations during the teams' work and the information obtained from the questionnaires and interviews, clearly showed that the second team could easily continue the project. This case study therefore showed that, in spite of the limitations imposed on the user interface structure (as described in section 3.2.3), the design teams found the system intuitive to use and could easily understand information entered by other team members or even previous teams. The context-rich structure and the ability of DiDeas II to reflect the design process and terminology that the team members are familiar with, is considered to be the key to the users' favourable response.

Some of the participants in the first case study were second year students who had just had been introduced to the design process in a course they were doing in the semester that the case studies were conducted. They commented that using DiDeas II helped them to understand the design process and how to apply it. This indicates that using DiDeas II in a company can help newly appointed design engineers to more easily adopt the company's own design style.

In the second case study, a comparison was done between distributed teams supported by DiDeas II and co-located teams working in a face-to-face situation. To reduce correlation effects in the case study, one team had to first use DiDeas II and then do the co-located work, while another team did the reverse. A significant difference observed between the two teams is that the team that first used DiDeas II, used the procedure and terminology that they learned there in the face-to-face project and worked more effectively and produced clearer documentation. The team that worked face-to-face before using DiDeas II produced fewer specifications and concepts in the face-to-face meeting than when they used DiDeas II, as well as the team that started by using DiDeas II. Both teams commented at the end of the case study that they worked more effectively using DiDeas II than without it. One important reason for this is that it was easier to handle hierarchical information using DiDeas II than using paper and pencil. When using DiDeas II, data can be added at any level of the system-subsystem tree. It was found from the questionnaires that all participants were satisfied that the tree view made it easy for them to access design information that was linked to particular system or subsystem. All of them agreed that the Project Edit interface could handle the information and could save and show the data in any subsystem.

The time spent by the team leader in each team in both the first and second case studies was recorded and then classified into four categories (Figure 7):

- Communication: e.g. when the team leader assigns tasks to different people tasks, or general non-project related communication with designers.
- Process information: reading and editing the design information such as creating or editing tree nodes (subsystems).
- Negotiation: e.g. explaining how and why one node or specification is created, problem identification and problem solving.
- Download/upload file: e.g. some sketch files to assist in negotiations.

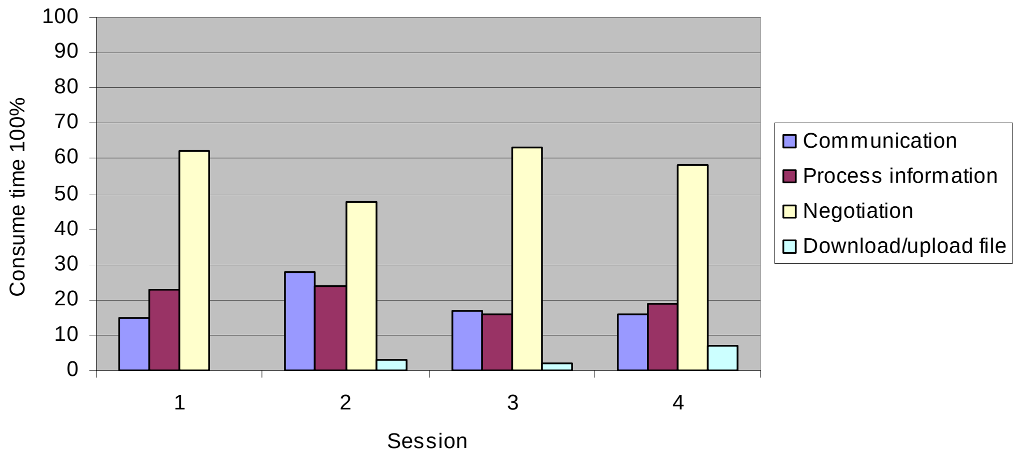

Figure 7: Analysis of Team Leader's Time

Sessions 1 and 2 in Figure 7 refer to the first case study, while sessions 3 and 4 refer to when DiDeas II was used in the second case study.

The results in Figure 7 were compared to a previous case study's results, where a team of six fourth year mechanical engineering students were observed in a system design project that they performed co-located, but without any design support software [18]. This showed that when using DiDeas II, the team leader required significantly less time for "Communication" and could spend a significantly larger proportion of his/her time on "Negotiation". Communication time using DiDeas II was reduced since the design information is clearly presented and easy for a designer to understand. Communication was found to be mainly concerned with data updating (due to different update intervals) and confirming file downloads. One aspect of Communication between team members, i.e. informing each other of the work done since the last meeting, consumed about 25% of meeting time in the previous case study, while when using DiDeas II, very little to no time was required for this activity. Negotiation was the major activity when using DiDeas II. This indicates that DiDeas II will help a team to spend less time on peripheral aspects and to spend more time considering concepts and thereby reaching a better result. This benefit, that one would expect to have with most design support systems, was retained in spite of the restrictions on the used interface imposed by keeping DiDeas II flexible.

4.2 Case Study Three

The third case study was aimed at evaluating to what extent DiDeas II can be customised in real industrial contexts. This case study was conducted with the assistance of three engineers from different small engineering companies.

The first engineer was from a small company designing and manufacturing cooking equipment for the canning industry. These devices are relatively simple from the design and process planning perspective. He reviewed DiDeas II with the ontology that was set up for the first two cases studies, but did not have the opportunity to actually adapt the ontology due to time constraints. However, he did conclude that DiDeas II can be adapted to his company's design style and that its use would be attractive for them since they currently have significant problems with design reuse, e.g. having to repeat much of the design process for modifications of current products because they do not have sufficient historical information just to consider the modifications, particularly when a one designer is doing a modification based on another designer's work, or where the previous design was done a long time ago. These problems could be solved by using a system such as DiDeas II. Further, due to its customisation properties, he even expected to be able to use DiDeas II as an aid in process planning.

The second and third engineers each worked in companies that do detail design and development of subsystems for other companies. Their companies do not do manufacturing in-house, but in some projects subcontract the manufacturing, while in other projects their designs are returned to the client for manufacturing. The two engineers independently reviewed DiDeas II, as it was set up for the previous case studies. From their comments, the following conclusions were drawn: Both engineers found the system-subsystem tree useful for placing the subsystem that they would typically work on, in its context. The both also indicated that the ability to customise DiDeas II to their companies' style of working, will make it a useful tool. The uses they saw for DiDeas II included aiding in communication directly between the design engineers and the project manager or client, since currently this communication has to be passed via a system engineer, which takes additional time. Even though it would take time for the design engineer to enter the relevant data into DiDeas II, both engineers indicated that it would be worth it if it can reduce the amount of time spent in meetings aimed specifically at reporting progress. In this role as communication channel, the ability to customise DiDeas II to reflect the design style of the particular company, client or project, was considered to be very important. One of the engineers also saw advantages in using DiDeas II as a training tool for novice designers.

Unfortunately, constraints on the availability of the design engineers in this case study, did not permit adapting DiDeas II according to their preferences and having them evaluate the changes. However, DiDeas II was adapted after the interviews to incorporate their comments, as shown in Figure 8.

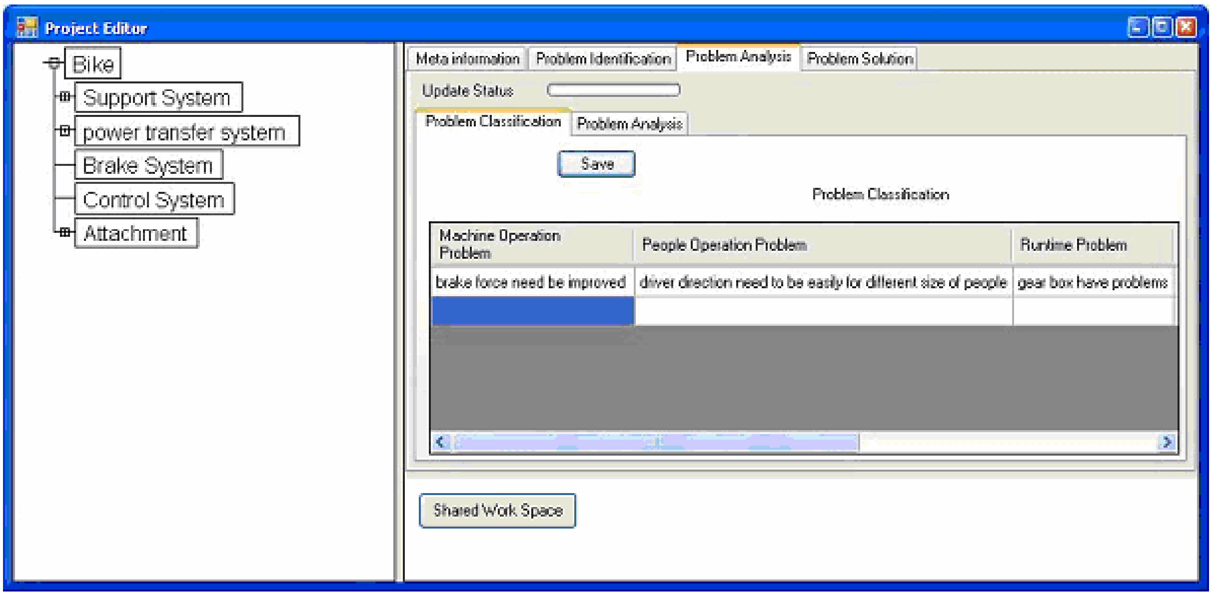

Figure 8: Project Edit Interface Revised after Case Study 3

5 CONCLUSIONS

The need exists for flexible (easily adaptable) design support tools for small and medium sized engineering companies. Flexibility in this context indicates that a given software tool can be adapted to a wide spectrum of design styles without requiring source code changes. DiDeas II, a design support system presented in this paper, which employs an ontology-based approach combined with the notions of elements and relations from conceptual graphs, aims at satisfying this need. DiDeas II allows changes to design terminology and procedures to be implemented by users of the software, without having to change the underlying source code.

DiDeas II is composed of two programs, one running on the server side and one running at each client (typically a member of the design team) respectively, which communicate using a TCP/IP connection. The server program contains the ontology editor (which sets the design style) and the database which manages the project data and the ontology data. The client side program provides a user interface to let users perform their design work by entering or editing the project's information. This user interface uses a tree view of the system-subsystem structure as the main means of navigating through the information. Tabbed pages are used to convey an implied (but not enforced) design procedure, and tables to manipulate many-to-many relationship sets. Most of the particulars of the user interface can be adapted by the ontology editor to suit the design process and terminology that the design team employs.

In the form used in the case studies presented in this paper, DiDeas II can handle specification development on different levels of the project system-subsystem structure, thereby helping to manage baselines. By using the tree view, tabs and tables, the design information in the early design phases can be displayed in a systematic and understandable manner.

This paper shows that an ontology-based approach can be used to create a design support tool that can be adapted for different design procedures and terminologies, without having to change the source code. The case studies presented here show that such a design support tool has the potential to significantly improve communication between team members, reduce time required for information exchange, facilitate distributed and asynchronous work and promote design re-use. They also confirmed that DiDeas II's ability to be adapted to the companies' or projects' design styles, is of critical importance in small enterprises.

The limited scope of the case studies precludes firm conclusions. They do, however, indicate that it is worth investing in the further development of DiDeas II to the stage where industries evaluate it in practical situations. The inclusion of functional analysis diagrams with hierarchies, loops and branches in an ontology-based approach is of particular research interest.

REFERENCES

[1] O'Grady, P., Pamers, D. & Bolsen, J., 1988, Artificial Intelligence Constraint Nets Applied to Design for Economic Manufacture. *Computer Integrated Manufacturing*, Vol. 1, pp. 204-210

[2] Basson, A.H., Bonnema, G.M. & Liu, Y., 2003, A Flexible Design Information System, *2003 International CIRP Design Seminar Laboratoire 3S*, Grenoble, France, May 12-14.

[3] Blanchard, B.S. en Fabrycky, W.J. "Systems Engineering and Analysis", 4th ed, Prentice Hall, London, 1998.

[4] Pahl, G. & Beitz, W., 1997, *Engineering Design – A System Approach*, 2nd ed., Springer, London.

[5] Ullman, G.D., 2003, *The Mechanical Design Process*, Chapman& Hall, New York.

[6] Ulrich, K.T. & Eppinger, S.D., 1995, *Product Design and Development*, McGraw-Hill, Inc., New York, U.S.

[7] Nigel, C., 1996, *Engineering Design Methods*, 2nd ed, John Wiley & Sons, Chichester.

[8] Vinney, J.E., Blount, G.N. & Noroozi, S., 1999, *A Conceptual Design ASsitant CODAS*, In *Proceeding of the 1999 CIRP International Design Seminar*, Enschede, the Netherlands, 24-26, March.

[9] Varma, A., Dong, A., Chidambaram, B., Agogino, A. & Wood, W., 2003, Web-based Tool by Engineering Design. *Journal of Integrated Design and Process Science*, Vol. 7, No. 3, pp. 95-108.

[10] Curry, A. & Stancich, L., 2000, The Intranet – an Intrinsic Natural Component of Strategic Information Management?. *International Journal of Information Management*, Vol. 20, pp. 249-268.

[11] Babu M. Joglekar N. Ganiji A. Ramani K., 2003, *Flexible Software Framework for Collaboration Systems, Collaborative Design Tools*, In *International CIRP Design Seminar*, Grenoble, France, May 12-14.

[12] Wang, L.H., Shen, W.M., Xie, H., Neelamkavil, J. & Pardasani, A., 2002, Collaborative Conceptual Design – State of the Art and Future Trends, *Computer-Aided Design* Vol. 34, pp. 981-996.

[13] Schueller, A., 2001, *Aspects of Distributed Conceptual Design Support*, Ph.D. Thesis, Stellenbosch University.

[14] Oscar, S., Mariano F. L.& Asunción G. P., Methodologies, tools and language for building ontology. Where is their meeting point? *Data & Knowledge Engineering* 46. 2003 P41 - 46

[15] Neches, R., Fikes, R.E., Finin, T., Gruber, T.R., Senator, T. & Swartout, W.R., 1991, Enabling Technology for Knowledge Sharing, *AI Magazine* 12 No.3 pp. 36–56.

[16] Sowa, J.F., 1992, *Conceptual graphs summary, Conceptual Structure,* in: Nagle, T.E., Nagle, J.A., Gerholz, L.L. & Eklund, P.W (Ed.), Current Research and Practice, Ellis Horwood Ltd., Chichester.

[17] Schueller, A. & Basson, A.H., 2001, *A Framework for Distributed Conceptual Design, Design Management - Process and Information Issues.* In *Proceedings of the 13th International Conference on Engineering Design ICED01*, Glasgow, U.K., pp. 385-391

[18] Liu, Y, 2007, *A Flexible Distributed Design Assistance Tool for Early Design Phases*, PhD Dissertation, Stellenbosch University, South Africa.

Contact: AH. Basson
Department of Mechanical and Mechatronic Engineering
Stellenbosch University
Private Bag X1
Matieland 7602
Stellenbosch
South Africa
Phone: +27 (021) 808-4250
Fax: +27 (021) 808-4958
e-mail: ahb@sun.ac.za