

---

## **Product development platform for real-time capture and reuse of evolving product information**

---

Amaresh Chakrabarti\*, Srinivas Kota,  
Nageshwar Rao and Sekhar Chowdary

Innovation, Design Study and Sustainability Laboratory (IdeasLab),  
Centre for Product Design and Manufacturing,  
Indian Institute of Science, Bangalore 560012, India  
E-mail: ac123@cpdm.iisc.ernet.in  
E-mail: srinivas@cpdm.iisc.ernet.in  
\*Corresponding author

**Abstract:** Product life cycle management promises management of all intellectual assets generated for all life cycle stages of a product. In engineering, over 75% of design activity comprises case-based design – reuse of previous design knowledge to address a new design problem. This means that rationale capture and reuse are critical in design and redesign projects. The goal of this paper is to report empirical study of designing carried out to understand the needs and constraints for design rationale capture, and to detail the concept, implementation and preliminary evaluation of an unobtrusive, real-time design rationale capture framework.

**Keywords:** design evolution; knowledge management; knowledge reuse; rationale capture; product data model; product life cycle management.

**Reference** to this paper should be made as follows: Chakrabarti, A., Kota, S., Rao, N. and Chowdary, S. (2007) 'Product development platform for real-time capture and reuse of evolving product information', *Int. J. Product Lifecycle Management*, Vol. 2, No. 3, pp.207–227.

**Biographical notes:** Amaresh Chakrabarti is an Associate Professor at the Centre for Product Design and Manufacturing, Indian Institute of Science, Bangalore, India. He did his PhD from the University of Cambridge, UK. His research interests are creativity and synthesis, eco-design, collaborative design, design research methodology and design for variety.

Srinivas Kota is a Research Scholar at the Centre for Product Design and Manufacturing, Indian Institute of Science, Bangalore, India. He received an ME degree from the BITS, Pilani. His research interests are eco-design, conceptual design, sustainability, life cycle thinking and product design.

Nageshwar Rao is a Project Assistant at the Centre for Product Design and Manufacturing at Indian Institute of Science, Bangalore, India. He has a BE degree in Mechanical Engineering from the Kuvempu University, Karnataka, India.

Sekhar Chaowdary is a Project Associate at the Centre for Product Design and Manufacturing at Indian Institute of Science, Bangalore, India. He has an ME in Mechanical Engineering from Indian Institute of Science, Bangalore, India.

## 1 Introduction

Product life cycle management promises management of all intellectual assets generated for all life cycle stages of a product (Fenves et al., 2003). This includes supporting capture and structure of information generated about an evolving product during its development process (Fenves et al., 2003). In engineering, it is conservatively estimated that more than 75% of design activity comprises case-based design – reuse of previous design knowledge to address a new design problem (Ullman, 1997). Many structures for product information (Ullman, 1989; McGinnis and Ullman, 1992; Blessing, 1994; Chakrabarti and Blich, 1994) and rationale information (Lee, 1991; Ullman, 1991) have been suggested; some have been turned into a tool, e.g. described in Lee (1991) and Ullman (1991). However, a tool for automated, real-time capture of structured product information including rationale information, especially as a product evolves through the development process, without having to interrupt the designer, has yet to be developed. The primary reason for this seems to be the mismatch between the speed of the problem solving cycle and that of its capture (Ullman, 1991). Retrospective tools are variously unreliable, for reasons including bias, rationalisation and forgetfulness (Minneman, 1991); appropriate tools must be developed for structure, capture and re-play of design information (Kuffner and Ullman, 1991). Consequently, some have suggested the use of annotated video information as a record and rationale of the proceedings during a product development process (Tang, 1989; Minneman, 1991).

We feel that a middle ground is needed between complete structuring of product information (good for reuse but effort-intensive and less reliable to create) and basic video information without much structuring (difficult for reuse but not effort-intensive to create), where some structuring of the data happens because of the way the work is carried out, without hampering the flow of work, with scope of further rationalisation if time and effort is available. So, an in-between solution (neither fully structured, nor fully unstructured) is needed, which allows designing to be carried out at its usual speed, while capturing information in a semi-structured way.

The goal of this paper is to report empirical study of designing carried out to understand the needs and process constraints for design rationale capture, and to describe a rationale capture framework, and its preliminary evaluation, developed to support capture of rationale.

## 2 Current practice

In current practice, there is not enough information recorded to answer all the questions raised during communication of design and for redesign support (Ullman, 1991). The information available within current Computer-Aided Design (CAD) tools is not sufficient to know the rationale of product development. At best, we have the requirements on one side and the final design drawings (with some explanation) on the other side of the process. Evolution of the product, design communication sessions in between and their rationale are not stored. While there are a number of methods and tools available for capturing a design process and its rationale, they all lack in something or the other, there is no tool available to support capture of all the information needed by a designer. Also, not all information can be represented in sketches or drawings (e.g. cost evaluation) and are currently expressed as utterances by a designer.

There are different approaches for capturing information and rationale during design, such as designer's notebook, note taking by a design historian, computer tools based on segmentation models like gIBIS (graphical Issue-Based Information System), video recording of designing and interviewing of the designer. We compared these approaches against the following criteria:

- Is information captured in a structured form?
- Is information captured detailed enough for understanding by a re-designer?
- Is any extra effort needed to structure the information?
- Are all information and rationale generated in the process captured?
- Does capture take place in real-time?

These questions are important in the context of the time and effort required for storing and reusing design rationale. Table 1 summarises our comparison of these approaches. The answers in capitals and within brackets under headings of each column specify the desired result from the evaluation used in that column.

**Table 1** Comparison of alternative rationale capture approaches

	<i>Structured</i> (YES)	<i>Information detail</i> (YES)	<i>Extra effort</i> (NO)	<i>All information captured</i> (YES)	<i>Real-time</i> (YES)
Designer's notebook	No	No	Yes	No	Yes
Design historian	Yes	No	Yes	No	Yes
gIBIS	Yes	Yes	Yes	No	No
Video recording	No	No	Yes	Yes	Yes
Interviewing	Yes	Yes	Yes	No	No
Video + segmentation	Yes	Yes	No	Yes	Yes

An analysis done on a protocol study on redesign of an already designed product (Kuffner and Ullman, 1991) found the following percentage of questions asked by designers on the various aspects of information and rationale; 47% questions towards the construction of components, assemblies, interfaces and features, 22% questions towards their location, 20% towards their operation and 11% towards their purpose. An ideal design rationale capture system should capture details about features, components, assemblies and relations between them with the intent behind creating those. It should capture the information in real-time without extra effort from the designer and others as the design process proceeds.

### 3 Overall idea, objectives and methodology

#### 3.1 Overall idea

The overall idea proposed here is to develop a product design platform which would capture the evolving product information automatically and provide links to browse and reuse the same without extra effort from the designer.

### 3.2 Objectives

The main objectives are to develop a platform

- for designers to explore and create product geometry, and be supported in terms of the product evolution through a real-time version tree with snapshots of the structure of the product after each conceivable steps of change to the product
- to create automatically the product structure with parts and relationships for each snapshot
- to capture automatically an audio–video record of the product development process carried out by the designer
- to divide the captured audio–video record into clips related to the proceedings between every two snapshots of the product structure.

### 3.3 Methodology

In order to identify the characteristics of product information in different stages of the design, several design processes are video taped and analysed using protocol study methods (see Sections 4.1 and 4.2). Based on this analysis, a structure for an evolving product and its versions, and a framework for the intended support are developed (see Section 4.3). Implementation of these on software is discussed in Section 5. Evaluation of efficiency of design and rationale capture are done by asking designers and potential users of rationale to use the platform and giving feedback on effort and benefit of using the system for rationale capture and reuse (see Section 6).

## 4 Analysis

Three design experiments are conducted to understand how product structure evolves through a product development process, and what actions are performed by a designer. In the first experiment, one designer was used, who developed solutions (bill of materials and engineering drawings) to Problem 1 (see Appendix A) using pen, paper and traditional drawing tools. In the second experiment, one designer was used, who solved Problem 2 (see Appendix A) using pen, paper and a computer aided modelling package as tools. In the third experiment, one designer was used, who was given only a computer aided modelling package for use in solving Problem 3 (see Appendix A). All three experiments were video recorded and analysed using protocol study methods. Conversion of video (about 8 hours) in digital form and time-stamped transcription (about 3000 sentences) of its protocols led to identification of the following kinds of activities and information typically generated in the various stages of product development.

### 4.1 Activities performed by the designer

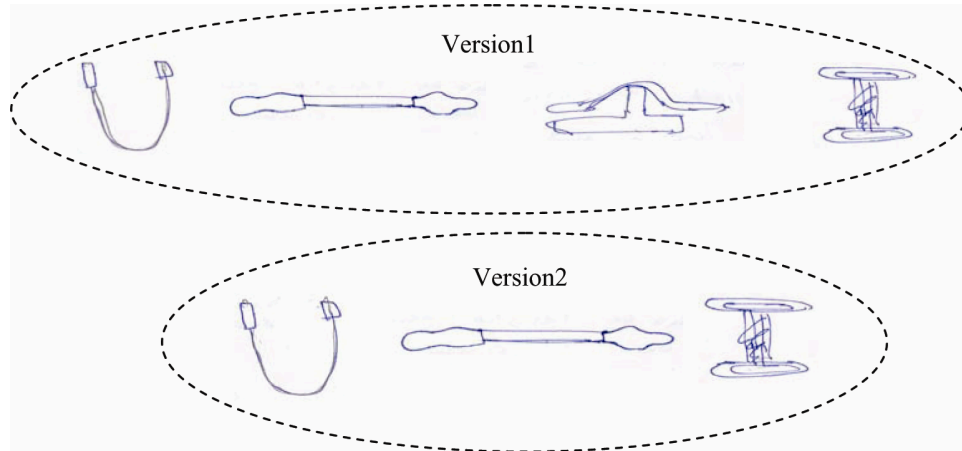
When a designer used pen and paper, he first wrote down the understood requirements and then tried to develop solutions by generating and evaluating a number of concepts. There were a large number of activities performed, such as evaluation of requirement satisfaction that were not recorded using pen and paper or the current computer assisted

modelling tools but only uttered while designing. Typical activities followed by a designer during designing that must be taken into account for developing a tool are identified below:

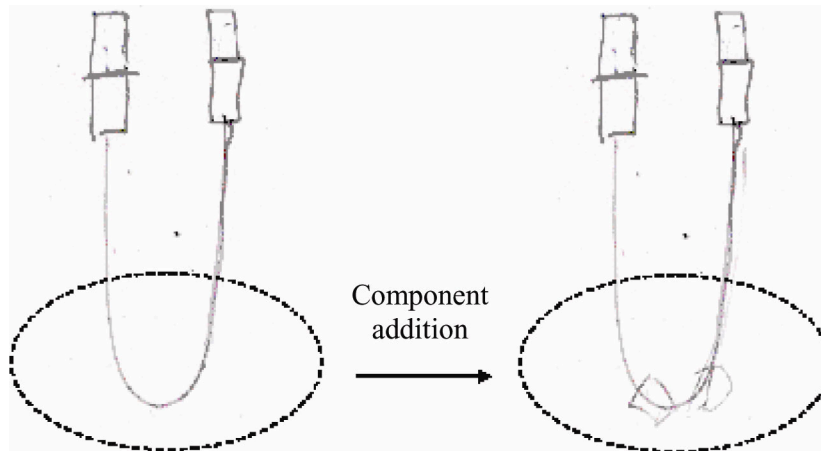
- 1 *Product version definition.* It is the specification of a concept. For example, in Experiment 2, the designer drew four sketches first and then said that these together constitute his first version of the product. After modifying and deleting some of these sketches and evaluating them, he reduced these to three assemblies and said this was his second version. Figure 1 shows the version definitions as sketched by the designer.
- 2 *Addition and subtraction of physical objects/information.* This entails addition or removal of components or features from an existing assembly or component. For example, first the designer drew a skipping rope and then to this he added two foot-clamps (see Figure 2). This figure shows the activity of adding components to an earlier assembly. Figure 3 shows the activity of material addition to a component.
- 3 *Addition and subtraction of relationships between objects.* In this activity, relationships between objects are specified or removed. For instance, the designer in Figure 4 first drew the two boxes attached without specifying any relationship between them (figure, left). He then added the detail of how the components were exactly related (right of the figure). Figure 4 shows an activity of addition of relation (thread) between two parts of the handle assembly.
- 4 *Substitution of object/information.* This is a combination of two activities; subtraction of already available object/information and addition of new object/information. For example, in a single activity, the designer removed the rope and modified the handle. Figure 5 shows the substitution of an object (rope).
- 5 *Focus to object or information.* In this activity, a designer concentrates on a particular object or information. For example, while designing workout equipment for executives, the designer drew a sketch representing a skipping rope with handles. In the next sketch, he drew only the handles without drawing the rope because he wanted to focus only on the handle. Figure 6 shows this focus activity.
- 6 *Defocus from object or information.* Here a designer defocuses, from a focused object or information, by representing the outline. For example, in the defocus activity shown in Figure 7, the designer sketched the details of the handle and then the outline of the handle.
- 7 *Change of the view or focus.* This activity is a combination of two activities; defocusing from the already focused object/information and focusing on others. For example in Figure 8, the designer was initially interested in the internal object (spring) within a rope assembly. Afterwards he changed his point of interest to the outside object (casing).
- 8 *Change of orientation of the objects.* Here, a given object is orientated in a different way as a result of the activity. For example, the designer in Figure 9 initially sketched the object vertically and then changed this to be horizontal.

There are some activities that are spoken only, and cannot be represented using drawings or as associations between objects and information. There should be some mechanism for capturing these activities, while allowing a designer to do the activities fast and with ease.

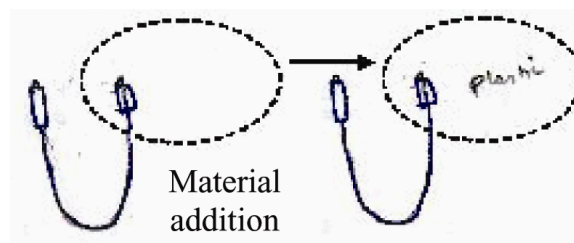
**Figure 1** Version definition



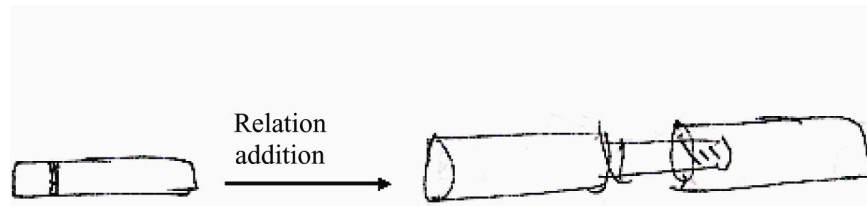
**Figure 2** Component addition to assembly



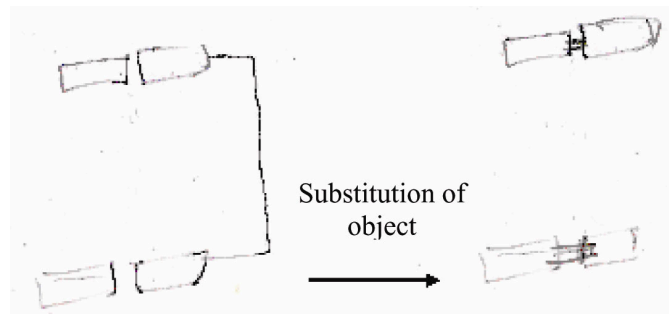
**Figure 3** Material addition



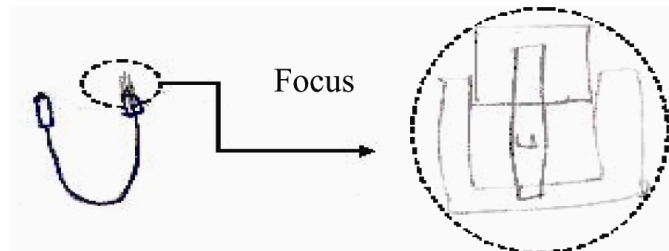
**Figure 4** Relation addition



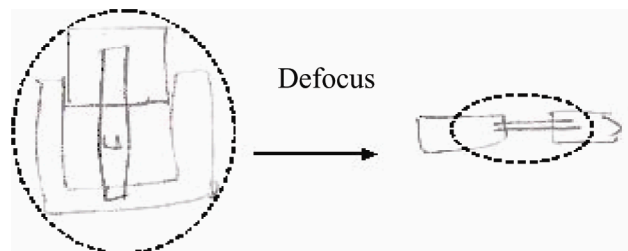
**Figure 5** Substitution of objects



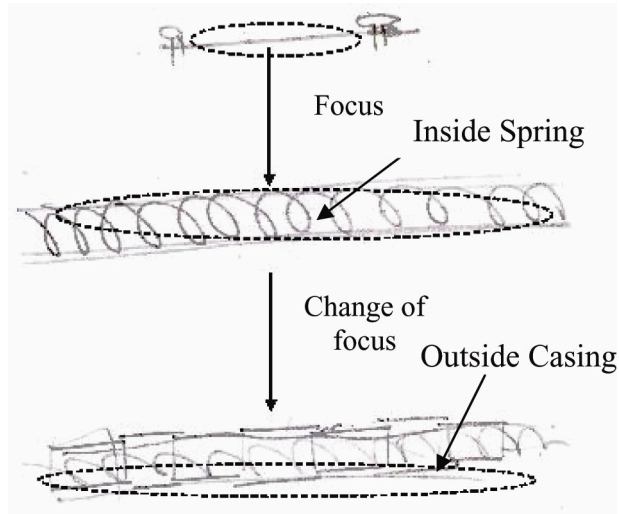
**Figure 6** Focus to object



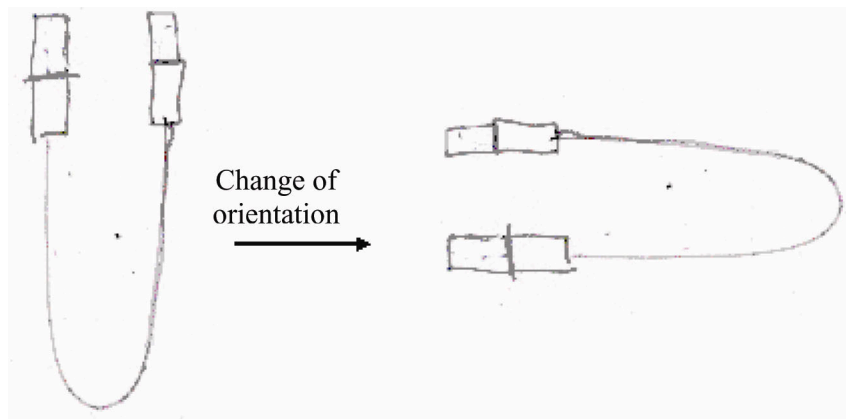
**Figure 7** Defocus from object



**Figure 8** Change of focus



**Figure 9** Object rotation



#### 4.2 Design process

The following are the broad design stages present in the design processes observed in the above experiments.

- 1 *0–15% of design time.* This is the initial stage in which given requirements of the design are studied, clarified and written down. A portion of transcription of design utterances in this stage is given in Figure 10.

Here, the first column shows the starting time, the second column shows the designer identification and the third column shows a transcription of the related audio. As can be seen from the utterances, the designer is trying to understand the problem given, by identifying the constraints and defining a problem statement.



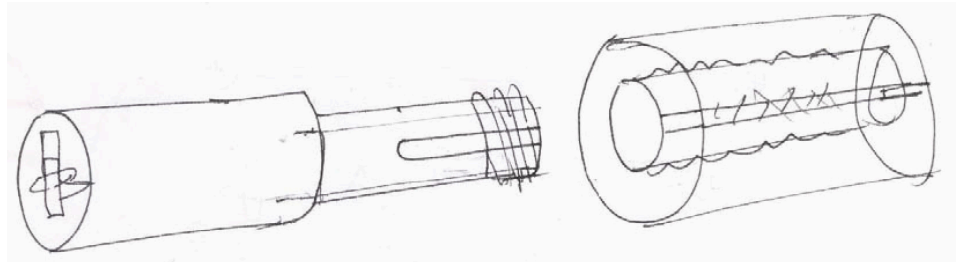
- 2 *15–40% of design time.* In this stage, ideas, spatial layouts and sub-assemblies of the design are specified. Figure 11 shows a transcription of a portion of the design process within this stage. Figure 12 shows a sketch of a component ‘handle’ during the conceptual and early embodiment stage.
- 3 *40–80% of design time.* In this stage, the interface details in the sub-assemblies are specified. Figure 13 shows a sketch of the component ‘handle’ in this stage.
- 4 *80–100% of design time.* In this stage, detailed dimensions, materials and manufacturing tolerances are specified. Figure 14 shows the detailed drawings of the ‘handle’ during the detailed design stage.

**Figure 10** A transcription of some utterances in the initial 0–15% time of design in a design experiment

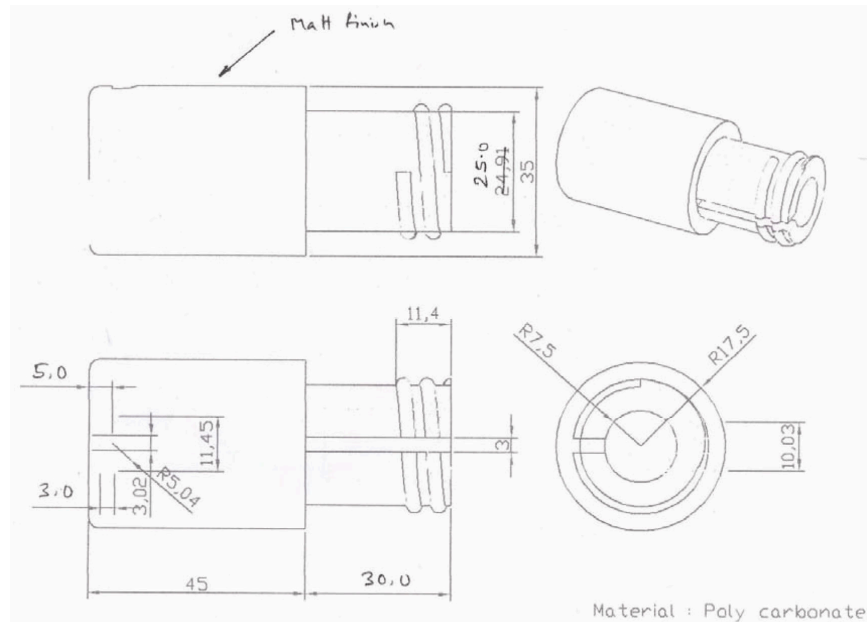
6.32	S1	First the problem is that executives requires exercises
6.45	S1	and why they require we don't know
6.48	S1	and because they are busy
6.57	S1	and they are reluctant to spend money to buy expensive gym equipment for personal use
7.01	S1	so they are miser
7.04	S1	miser, do not want to spend money for equipment
7.17	S1	which means that the assumption is they know that they have to do some exercise but they are not doing
7.23	S1	and they think that gymnasium equipment is very expensive and I don't need to buy
7.3	S1	because they are quite expensive probably
7.34	S1	well there are some personal use equipment available but they are expensive, ok
7.4	S1	this is a fact
7.44	S1	privacy is not there in gymnasium
7.52	S1	privacy is not there means lot of people are feeling, feeling what
8.03	S1	feeling shy of going there, body building exercise probably
8.09	S1	why they don't do exercise wearing the full dress (smile) strange
8.15	S1	current equipment occupies lot of space ok
8.28	S1	and usually are not portable
8.34	S1	these are problems with current equipment
8.39	S1	so the requirements are external requirements, some are constraints
8.46	S1	apart from the solution of the problem, requirements are that
8.52	S1	it should be easily setupable
9.07	S1	it should be setup easily and portable
9.12	S1	and should help in complete workout of the body
9.27	S1	ya
9.29	S1	first of all the thing is that whether we really achieving that

**Figure 11** Transcription for a portion of the 15–40% time of design

10.44	S1	First of all the gymnasium equipment is not required at all I feel
		may be I am thinking what is solution, I can give a chart where a chart be like you do these exercises and get rid of this problem you do these exercises and free hand exercises that is enough jogging you do some some some type of wrestling not wrestling some some type of yoga yogic exercises hata
10.49	S1	yoga it is called and all problem will be solved
11.17	S1	but see this problem I will think of the problem so
11.23	S1	I will write down problem solution
11.32	S1	one is
11.35	S1	one is manual for yoga and this can be customized
		may be I can have a software where you can give problem what are problems you have and you get some output with postures animation and you give some ways to do it and it is proved by many rishis and thousand years of research more that that yoga is very good for healthy and body and not this we are going to copy form british and american guys that we have to do with some rod put springs against it and do some strange exercises which has no meaning after all and we have to wait for one month to see some output I think it has not proved that by doing this exercises your heart is good your teeth are good means your health is good
11.48	S1	
12.4	S1	but if I see lot of yogic exercises especially we see madhurasan, we see bakasan we see what else lot of others is there first of all manually is a good option
13.02	S1	now let us think in the other way what can be done

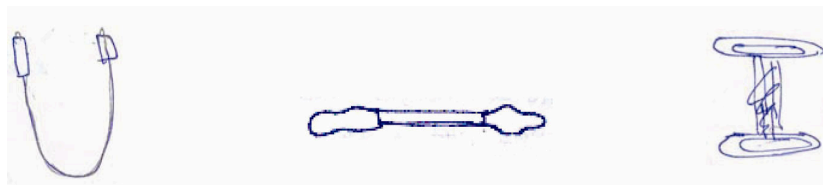
**Figure 12** Sketch of handle during the 15–40% time of design**Figure 13** Sketch of the handle in the 40–80% time of design

**Figure 14** Final drawing of the handle in the 80–100% time of design



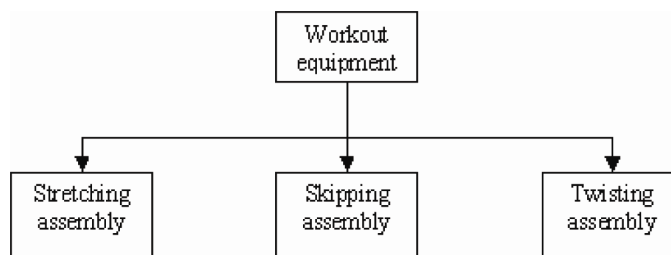
The definition of the assemblies delineating the product evolves throughout the design process. For example, in Experiment 2, the product designed is a personal workout equipment and initially consists of three different assemblies; the skipping assembly, the twisting assembly and the stretching assembly as shown in Figure 15 (left to right).

**Figure 15** Different assemblies of product



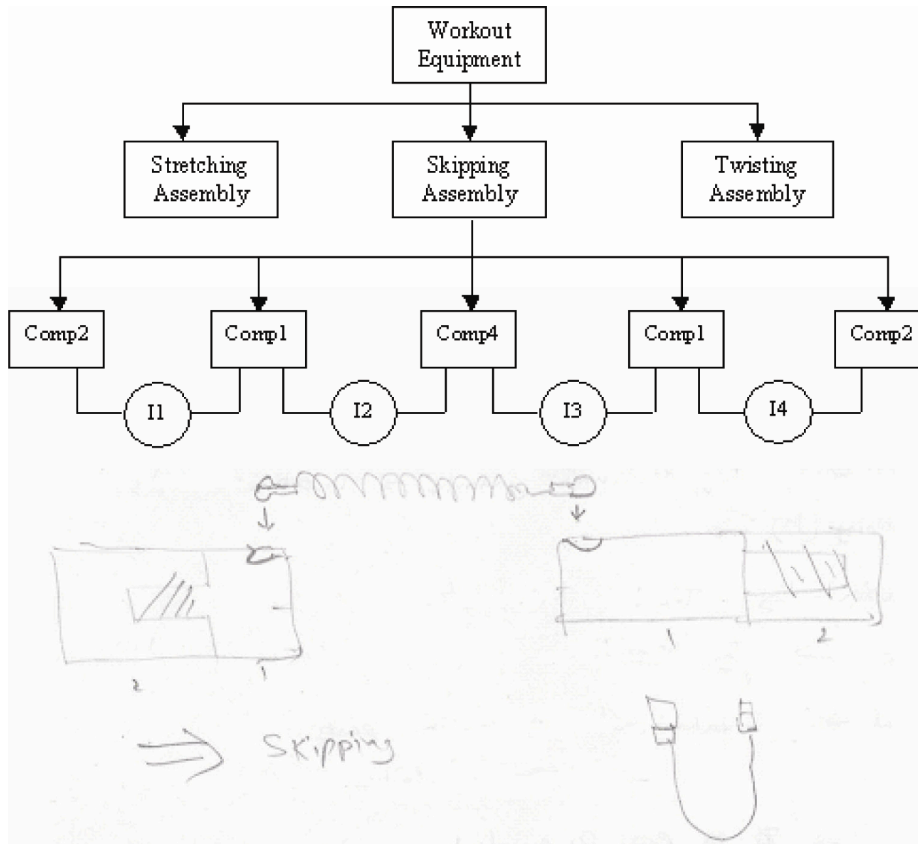
At this stage of design, the product configuration contains information about the main subassemblies of the product as shown in Figure 16.

**Figure 16** Workout equipment product configuration



The sub assemblies are subsequently detailed to consist of components, which have of features and relations between them. Figure 17 shows the product configuration at a more detailed stage with components and relations.

Figure 17 Product configuration with components and relations



An *assembly* is defined here as a collection of assemblies, components and relations between them. A *component* is an individual physical object in a product. *Features* are the characteristics of assemblies, components and relations. *Relations* are the connections among assemblies, components and features.

By looking at the product configuration at the top level, one should be able to identify and explore different assemblies, components belonging to respective assemblies and the relations between them.

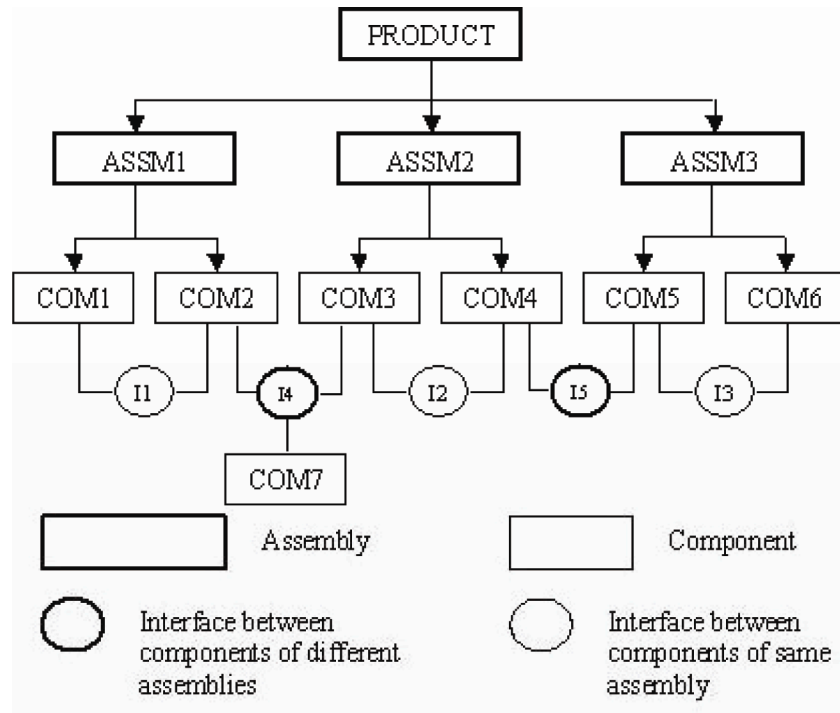
### 4.3 Framework

The findings in the previous sections are used to develop a suitable product model schema and a framework for real-time capture and reuse of evolving product information. The framework consists of the following entities: product structure, snaps, events, versions, version tree and audio–video clips. These components are discussed in detail below.

### 4.3.1 Product structure

A *product structure* is defined as an assembly of components (with features) and relationships. The product structure should be constructed automatically by extracting the information from the CAD package used by the designer as the designer performs modelling in the CAD package. Figure 18 shows a template of a product structure automatically created as a designer performs a design task.

**Figure 18** Product structure



Opening an assembly should display assembly properties such as assembly process, components belonging to the assembly, etc. Opening a component should display component properties such as mass, volume, surface finish, manufacturing process etc. As an interface is opened, interface properties such as the type of interface, the component features involved in the interfaces etc., should be displayed.

### 4.3.2 Event

An *event* is defined as any change made to the form, material or process, and have the duration between two consecutive snapshots, or calls to cost analysis/environmental impact analysis etc. and revisits to earlier snapshots.

### 4.3.3 Snap

A *snap* is defined as a snapshot of the structure of a product after an event. A snap should be created whenever

- an assembly is added/deleted/changed
- a component is added/deleted/changed
- a feature is added/deleted/changed
- a material is added/deleted/changed
- a manufacturing process is added/deleted/changed
- an assembly process is added/deleted/changed
- a visit is made to a previous snap
- a call/request is made for analysis.

We found that a designer often revisits the already created snaps.

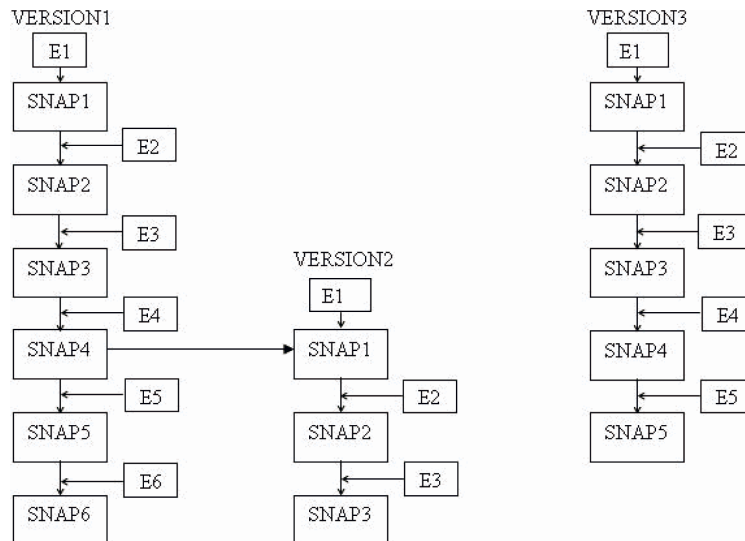
### 4.3.4 Version

A *version* is defined as a product structure that is stored under a separate version name.

### 4.3.5 Version tree

A *version tree* has a chronologically ordered series of versions, each with an ordered series of snaps with video clips for events in between. If a user wants to use a current snap to create new snaps, she should copy the snap to the current workspace and modify it using the activities listed above. Figure 19 shows the concept of a version-tree containing versions, snaps and events.

**Figure 19** Version-tree with versions, snaps and events



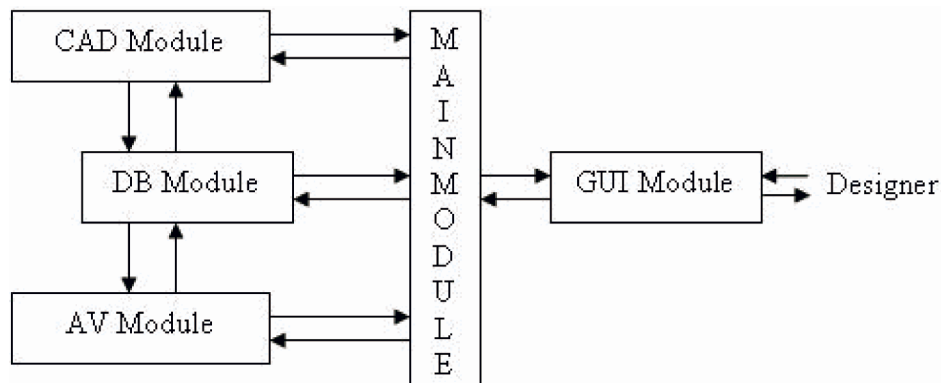
#### 4.3.6 Audio–video clips for an event

All the proceedings between the current and the preceding snaps are captured through audio–video recording, cut automatically into a video file, saved in an appropriate location, and a pointer to this is added to the version tree at the appropriate place between the two relevant snaps. Whenever a designer wants to see what happened during this event, she can go to that particular event clip and see the proceedings.

## 5 Implementation

In this preliminary version, we have concentrated on the development of the framework, and the core modules with important features. The work completed and future work remaining are discussed in Section 6. The overview of the implemented prototype is shown in Figure 20.

**Figure 20** Overview of implemented prototype



The prototype consists of the following five modules

- a MAIN Module
- b CAD Module
- c AV Module
- d DB Module
- e GUI Module

*MAIN Module:* It is the programme that integrates all the other modules. It interacts with the CAD module, the Audio–Video module (AV module), the Data Base module (DB module) and the Graphical User Interface (GUI) module. It is implemented in Microsoft Windows® environment using Microsoft Visual C++® language.

*CAD Module:* Modelling of the product is done here. It sends product information to text files. It is currently implemented in UniGraphics® in Microsoft Windows® environment using UGOpenFunc API®.

*AV Module:* It is the programme developed to automatically capture, cut and store the AV files. It is implemented in windows environment using DirectX® SDK and Microsoft Visual C++®.

*DB Module:* It is the database developed during the running of the programme. In this database all details of the CAD data pertaining to the versions, snaps and events are stored. It is implemented in Microsoft Windows® environment using Microsoft Visual C++®.

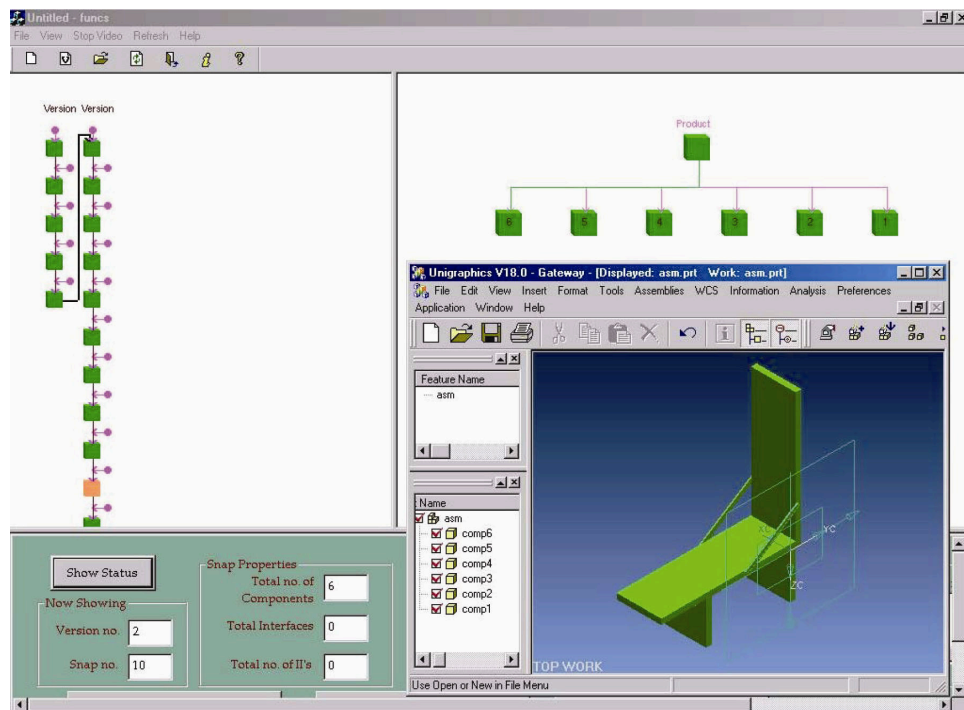
*GUI Module:* It is the interface between the main programme and the user. It is implemented in Microsoft Windows® environment using OpenGL® and Microsoft Visual C++®.

## 6 Example

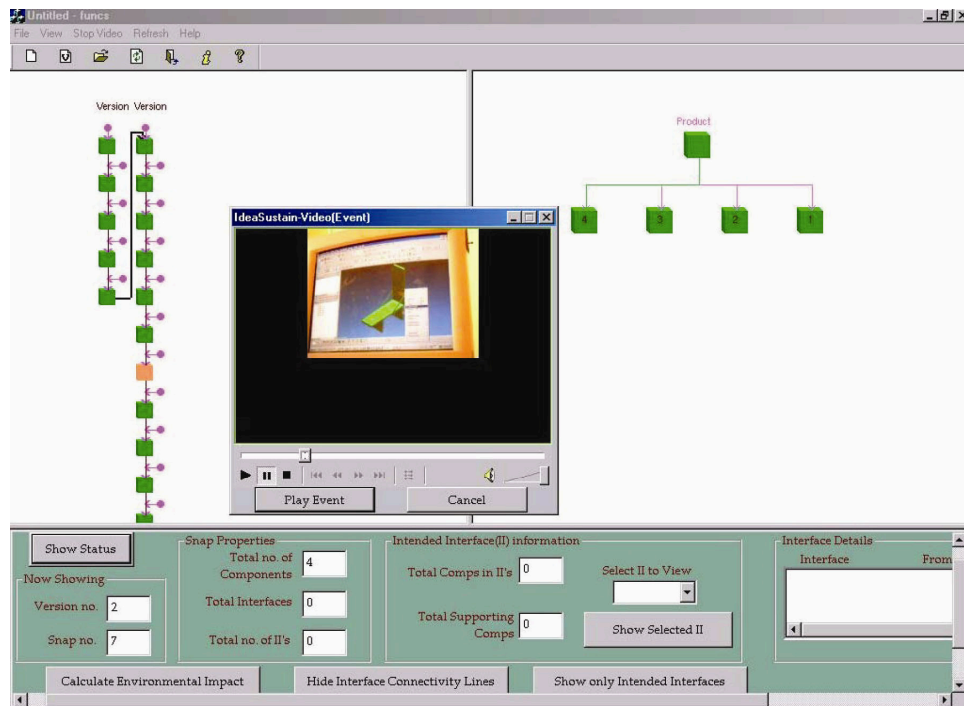
The following screen dumps (Figures 21–22) of the software developed with patent-pending – called *Idea-Sustain* (Chakrabarti and Kota, 2006) – are used to explain the functioning of the system. There are three main functions that are performed within the system:

- 1 creation and modification of a 3-dimensional assembly
- 2 exploring the details of the version tree and product structure
- 3 exploring the events via the video/audio clips attached to the version tree.

**Figure 21** Exploration of product structure and version tree on the software





**Figure 22** Exploration of an event using the software

We need a camera to be attached to the system where the designing and modelling is going on and it has to be initialised at the beginning of the work and is done through the steps which will be displayed when we start the programme. After that the programme takes care of capturing, cutting and storing the AV clips at appropriate positions using the logic developed for the activities performed by the designer. The designer has to work *thinking aloud* if designing alone or when ever he feels important information is generated and must be recorded, so that the event is captured through audio.

The first function is performed by calling a commercial CAD software and working within the software. The role of *Idea-Sustain* is to track the evolution of the product versions and the structure of the product created as a result.

The second function – of exploring the version tree and associated product structures is supported with an interface that provides a chronological as well as causal list of snaps on the left window (see Figure 22), and the product structure corresponding to any snap within the tree (in this case the highlighted one) on the right window. Specific details about the snaps can be examined by clicking on required component or assembly (to see geometric information) or by queries using the bottom window.

The third function – exploration of events – is performed using event information captured during the design process. The dots between snaps in the version tree are active markers for events – audio/video clips related to events constituted between the snaps immediately before and after the clip.

## 7 Evaluation, discussion and future work

The following set of questions was used to understand the ease of use of the software by testing it using two designers and six users. The numbers in the bracket indicate how many people made this point:

- Do you feel any difference between working with the software and working with conventional documents captured during design (e.g. engineering drawings, final CAD files, etc.):
  - *Designer feedback.* The main difference from conventional documentation is in the availability of the product structure/version tree and video clips (2/2). The concept of recording is good, particularly since evolution of each part can be traced (1/2). This makes learning from the scratch for a new user easy (1/2). Availability of voice recording is especially useful (2/2).
  - *User feedback.* The proposed system is much better than going through conventional documents (1/6). One can know how the product evolved (1/6).
- Do you see any difference between this environment and conventional CAD environment:
  - *Designer feedback.* Being able to modify from previous snaps is very useful (1/2). The facility to save the product evolution as events at the background is not available in conventional CAD systems (2/2). The tree structure gives a useful visual representation of the evolving product information (1/2). Modelling environment is the same (1/2).
  - *User feedback.* Conventional systems provide no explanation of designers' views unlike this (1/6). You can see the process from start to end (1/6). Construction (of the product and its procedure) is very clear in the proposed system, especially using audio/video clips (3/6). While recent history can be played back on CAD systems, being able to save different versions is novel and useful (1/6). Modelling environment is the same (3/6).
- What effort is required in structuring product development information:
  - *Designer feedback.* No extra effort is necessary for anyone already familiar with the CAD system (2/2).
  - *User feedback.* No structuring is necessary to see the information (5/6).
- What effort is required in exploring product development information:
  - *Designer feedback.* It is not easy to know where the next part is added without going through the next snaps exhaustively (1/2). Rectangular blocks being used for product structure give no semantic information about the nature of the parts/assembly (1/2).
  - *User feedback.* It is not easy to know where the next part is added without going through the next snaps exhaustively (1/6). Some videos are too long; it is better to have very small video clips (1/6). Version tree allows easy tracking of changes (1/6). Zooming facility is useful (1/6). It is very easy to navigate and explore (2/6).

To summarise, Idea-Sustain framework presented above does the following things:

- It automatically creates a product structure as a designer uses the CAD package for modelling.
- Each change made to the product is automatically saved as a distinct snap of the product structure so that all steps followed in the design process are available for exploration and reuse.
- It automatically captures the video with audio, cuts it and places in appropriate places for browsing.
- It does not explicitly chunk the rationale information, not even the requirements information, but all these are contained in the video clips (if there is anything mentioned, as to why change between relevant snaps happened), and development between snaps (i.e. change in product structure) tells what has happened. However, these can be further annotated if a mechanism (such as manager to do this) is found.

Currently, the implementation has the following limitations. Firstly, not all snap categories identified in Section 3.3 has been implemented – currently only product structure-related changes are captured. Secondly, the very early (e.g. sketching/problem understanding) and late processes (usage and after-usage scenarios) are currently not captured. Thirdly, the difference between intra-assembly relations and inter-assembly relations are currently not shown in the GUI. No detailed rationale-partitioning is currently available, and if a designer does not speak or do anything that can be recorded in audio/video clips, little rationale will be there (except the changes made in the product structure and corresponding steps taken on the computer) to be captured. Also, the evolution of requirements is currently not captured in an explicit sense. Finally, evaluation is currently at the level of whether a real design process can be carried out and explained using this implementation, and not a comparative evaluation to see how this helps better understanding or redesigning than supported by conventional designing or rationale capture means. All these form areas of further work.

## **Acknowledgements**

We acknowledge the help extended by Prabir Sarkar, Santosh Jagtap, Harsha T., Leelavathamma, Vishal Singh, Bipin Ghosh and Tharakeshwara.

## **Appendix A**

Following are the three problems given to designers.

### *Problem 1*

India has a large number of people with transferable jobs. They need to shift frequently from one place to other (every 1–2 years). And often face problems transferring present types of furniture, which are bulky and heavy. It is not economical for them to buy furniture and sell it before shifting to the other place. This furniture occupies a lot of

space and this is an additional problem, since they live in small houses. It takes more time to pack the furniture and it damages during transport if it is not packed properly.

Your task is to design a portfolio of furniture, which will help in solving these problems. Set-up time and effort on the part of user should be minimal.

At the end of the design, you are expected to provide part and assembly drawings and a bill of materials, along with any other details necessary for production of the product. You have to consider all the life cycle phases of device/product.

### *Problem 2*

Many modern executives find it difficult to spend spare time from their busy schedule to go to the gym for workout. On the other hand, they are often reluctant to spend money on expensive gymnasium equipment for personal use. There are some personal-use equipments available but they are expensive. Privacy is not there in gymnasium. Current equipments occupy a lot of space and are usually not portable.

Your task is to design a product that will help in solving these problems. Users should be able to use it without any difficulty in setting up the equipment. It should be portable and should help in complete workout of the body.

At the end of the design process, you are expected to provide part and assembly drawings and a bill of materials, along with any other details necessary for production of the product. You need to consider all the life cycle phases of the product.

### *Problem 3*

We use different brooms to clean different areas in our house. To clean dry surfaces one type of broom and to clean wet surfaces another type of broom is used. We use cloth or sponge to remove water. To remove spider-nets, we use other types of broom. There is lot of manual effort involved in cleaning. A common problem in using many of these equipments is that they make air dusty in the area where cleaning is done, which requires additional time to clean. Also there is difficulty involved in cleaning corners using these equipments.

Your task is to design a product that will help in solving these problems. Users should be able to clean interior and exterior areas of a house, including windows, doors, floors and roofs. The product should remove dust, water and dirt accumulated in the corners as well. The product should require only one person to operate.

At the end of the design process, you are expected to provide part and assembly drawings and a bill of materials, along with any other details necessary for production of the product. You need to consider all the life cycle phases of the product.

## **References**

- Blessing, L.T.M. (1994) *A Process Based Approach to Computer Supported Engineering Design*. Enschede, The Netherlands: Thesis, University of Twente.
- Chakrabarti, A. and Bligh, T.P. (1994) 'An approach to functional synthesis of solutions in mechanical conceptual design. Part 1: Introduction and knowledge representation', *Research in Engineering Design*, Vol. 6, pp.127–141.
- Chakrabarti, A. and Kota, S. (2006) *Idea-Sustain*, Patent Application No. 858/CHE/2006, Patent Office, Government of India.

- Fenves, S.J., Sriram, R.D., Sudarsan, R. and Wang, F. (2003) 'A product information modeling framework for product lifecycle management', Paper presented in the Proceedings of the *PLM Symposium*, July 16–18, Bangalore, India.
- Lee, J. (1991) 'Extending the potts and bruns model for recording design rationale', Paper presented in the Proceedings of the *13th International Conference on Software Engineering*, May.
- Kuffner, T.A. and Ullman, D.G. (1991) 'The information requests of mechanical design engineers', *Design Studies*, Vol. 12, pp.42–50.
- McGinnis, B.D. and Ullman, D.G. (1992) 'The evolution of commitments in the design of a component', *ASME Journal of Mechanical Design*, Vol. 114, pp.1–7.
- Minneman, S.L. (1991) *The Social Construction of a Technical Reality: Empirical Studies of Group Engineering Design Practice*. Thesis: Dept. Mechanical Engineering, Stanford University.
- Tang, J.C. (1989) *Listing, DRAWING and Gesturing in Design: A Study of the Use of Shared Workspaces by Design Teams*. Thesis: Dept. Mechanical Engineering, Stanford University.
- Ullman, D.G. (1989) 'A taxonomy of mechanical design', *ASME International Conference on Design Theory and Methodology*, DTM89, Quebec.
- Ullman, D.G. (1991) 'Design histories: archiving the evolution of products', Paper presented in the Proceedings of the *DARPA workshop on Manufacturing*, February 5–6, Salt Lake City, Utah.
- Ullman, D.G. (1997) *The Mechanical Design Process* (2nd ed.). New York, NY: McGraw-Hill.