INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15 – 18, 2005

A DESIGNER WORKBENCH WITH REAL-TIME CAPTURE AND REUSE OF INFORMATION ON PRODUCT EVOLUTION AND RATIONALE

Amaresh Chakrabarti, Srinivas Kota, Nageshwar Rao, Sekhar Chowdary

Abstract

Product Life Cycle Management promises management of all intellectual assets generated for all life cycle phases of a product [1]. This includes supporting capture and structure of information generated about an evolving product during the PD process [1]. Many structures for product information [3, 4, 5, 6] and rationale information [7, 8] have been suggested, and some turned into a tool, e.g. [7, 8]. However, a practical tool for automated capture and structure of product information including rationale information without interrupting the working of the designer, especially as a product evolves through the PD process, is yet to be developed. The framework reported here has been developed by analysing the proceedings from design experiments and literature to capture, segregate and store the information generated in product design without interrupting designer. The main features of the framework are implemented into the preliminary version of a software tool and evaluated for the ease of use and reuse for designers and re-designers (re-users of rationale). The main functions currently supported are: creation and modification of a three-dimensional assembly, exploring the details of a version tree and product structure, and, exploring the events via video/audio clips attached to the version tree.

Keywords: *rationale capture, product data model, design evolution, product life cycle management, knowledge management*

1. Introduction

Product Life Cycle Management promises management of all intellectual assets generated for all life cycle stages of a product [1]. This includes supporting capture and structure of information generated about an evolving product during its development process [1]. In engineering, it is estimated that over 75% of design activity comprises case-based design – reuse of previous design knowledge to address a new design problem [2]. Many structures for product information [3, 4, 5, 6] and rationale information [7, 8] have been suggested; some turned into a tool, e.g., [7, 8]. However, a tool for automated, real-time capture of structured product information including rationale information without interrupting the designer, especially as a product evolves through the development process, 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. Retrospective tools are variously unreliable, for reasons including bias, rationalisation, and forgetfulness [9]; appropriate tools must be developed for capture, structure and re-play of design information [10]. Consequently, some suggest annotated video information as a record and rationale of product development [11].

We feel that a middle ground is needed between complete structuring of product information (good for reuse but effort-intensive and less reliable) and basic video information without much structuring (difficult for reuse but not effort-intensive), where some structuring of the data happens because of the way the work is carried out, without hampering the flow of work, with scope for further rationalisation if time and effort is available. So, an in-between solution (not fully structured, not fully unstructured) is needed, which allows design 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, based on these findings, development and evaluation of a preliminary version of a rationale capture framework.

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 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).

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.

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

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. YES for the first four questions and NO for the last question are the ideal scenarios.

Table 1. Comparison of Alternative Rationale Capture Approaches

	Complete (YES)	Structured (YES)	Information Detail (YES)	Real-time (YES)	Extra Effort (NO)
Designer's notebook	No	No	No	Yes	Yes
Design historian	No	Yes	No	Yes	Yes
gIBIS	No	Yes	Yes	No	Yes
Video recording	Yes	No	No	Yes	Yes
Interviewing	No	Yes	Yes	No	Yes
Video + Segmentation	Yes	Yes	Yes	Yes	No

An analysis done on a protocol study on redesign of an already designed product [10] 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 that 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 step of change to the product
- To automatically create the product structure with parts and relationships for each snapshot
- To automatically capture 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 4.1-4.2). 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 efficacy of design and rationale capture is done by asking designers to use the software platform developed to solve design problems, subsequently for a group of users of rationale (e.g., re-designers) to use the platform for developing an understanding of the above design decisions and their rationale, and then judging the relative quality of their understanding of the rationale vis-à-vis using conventional documentations (see Section 7).

4. Analysis

Two design experiments are conducted to understand how a product structure evolves through a product development process, and what actions are performed by designers. In the first experiment, one designer was used, who developed solutions (bill of materials and engineering drawings) to a problem using pen, paper and traditional drawing tools. In the second experiment, one designer was used, who solved another problem using pen, paper and a computer aided modelling package as tools. All these experiments are 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 in 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:

Product version definition: It is the specification of a concept. For example, in Experiment 2, the designer sketched 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.

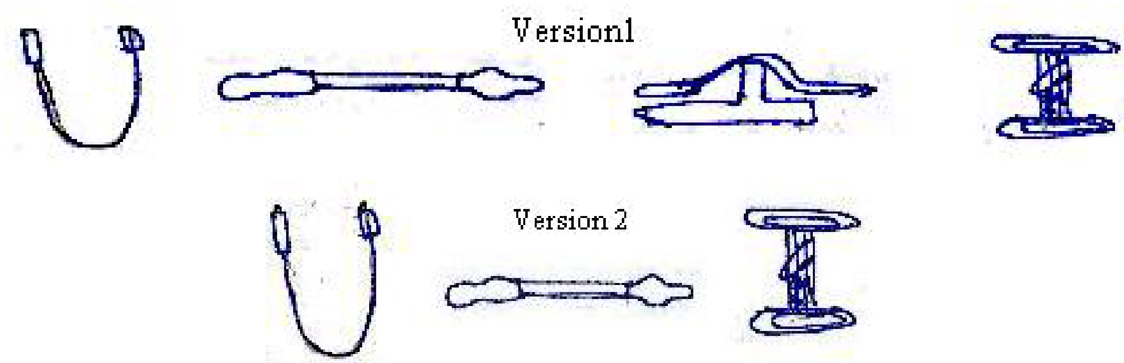

Figure 1. Version definition

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.

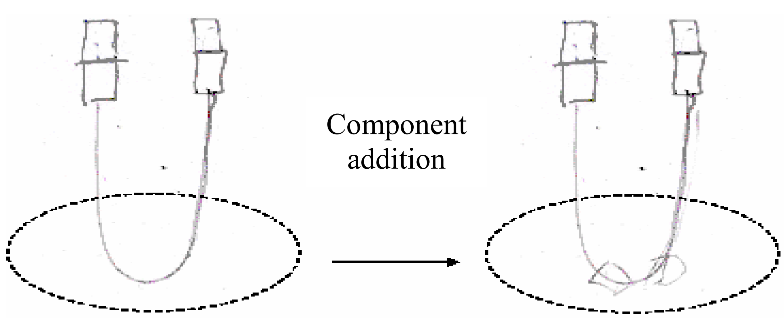

Figure 2. Component addition to assembly

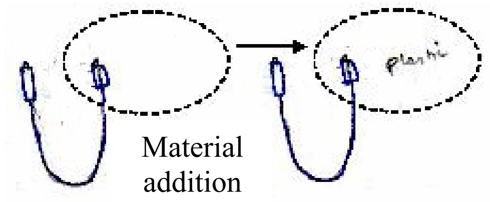

Figure 3. Material addition

Addition and subtraction of relationships between objects: In this activity, relationships between objects are specified or removed. For example, the designer in Figure 4 initially drew the two boxes attached without specifying any relationship between them (left of the figure). After this, he added the detail of how the components were exactly related (right of the figure). Figure 4 shows this activity of addition of relation (thread) between the two parts of the handle assembly.

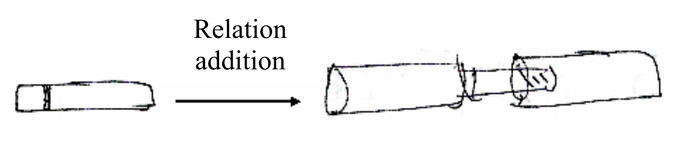

Figure 4. Relation addition

Substitution of object/information: This activity 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 part. Figure 5 shows the substitution of an object (rope).

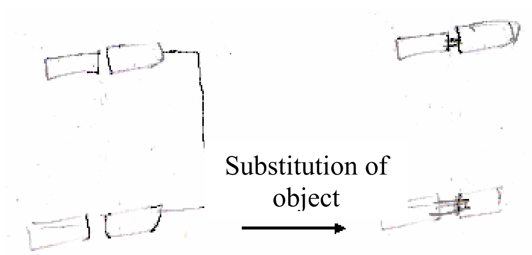

Figure 5. Substitution of objects

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 on the handle. Figure 6 shows this focus activity.

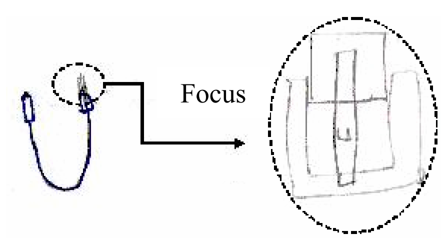

Figure 6. Focus to object

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 sketches the details of the handle and then the outline of the handle.

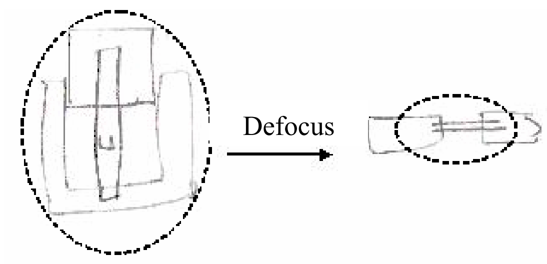

Figure 7. Defocus from object

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 on the internal object (spring) within a rope assembly. Afterwards he changed his point of interest to the outside object (casing)

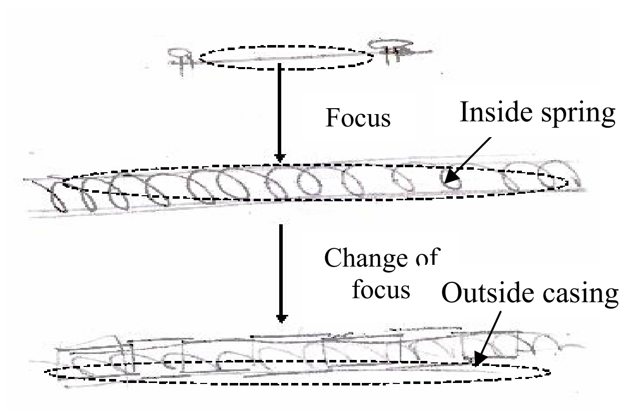

Figure 8. Change of focus

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.

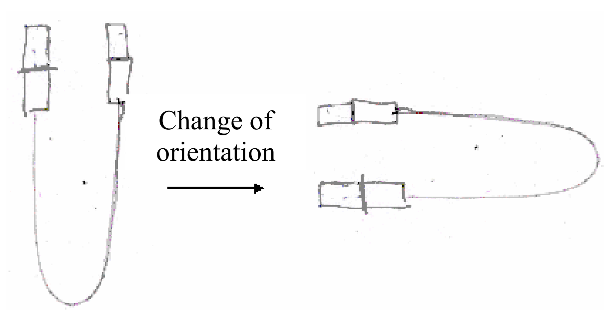

Figure 9. Object rotation

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

4.2 Design process

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

Task clarification: 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.

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 gymasium
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 constratints
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 10 A transcription of task clarification utterances in a design experiment

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 seen from the utterances, here the designer tries to understand the problem given by identifying the constraints and defining a problem statement.

Conceptual Design: 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 stage.

10.44	S1	First of all the gymnasium equipment is not required at all I feel
10.49	S1	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 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
11.48	S1	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
12.4	S1	but if I see lot of yogic exercises expecially 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 11 A transcription for a portion of the conceptual design stage

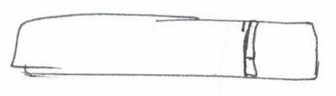

Figure 12 Sketch of Handle during the Conceptual Stage

Embodiment Design: In this stage, the interface details in the sub-assemblies are specified. Figure 13 shows a sketch of the component "handle" in this stage.

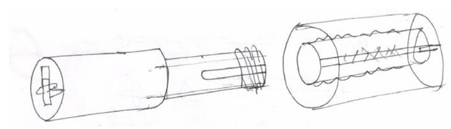

Figure 13 Sketch of the Handle in the Embodiment Stage

Detailed design: 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.

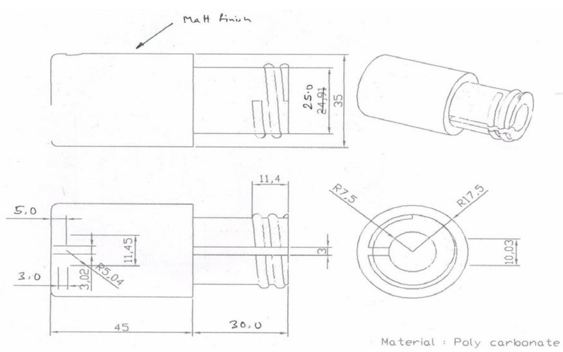

Figure 14 Final drawing of the handle in detailed design

The definition of the assemblies delineating the product evolves throughout the design process. For example, in Experiment 2, the product designed is 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).

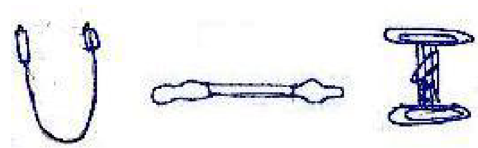

Figure 15. Different assemblies of Product

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

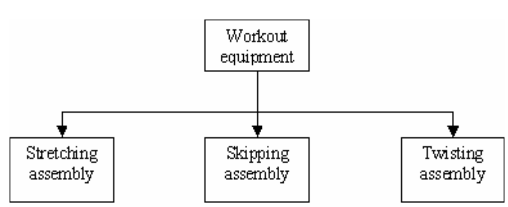

Figure 16. Workout Equipment Product Configuration

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

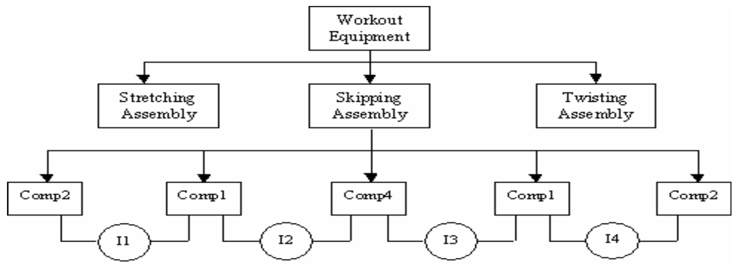

Figure 17. Product configuration with components and relations

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 between 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.

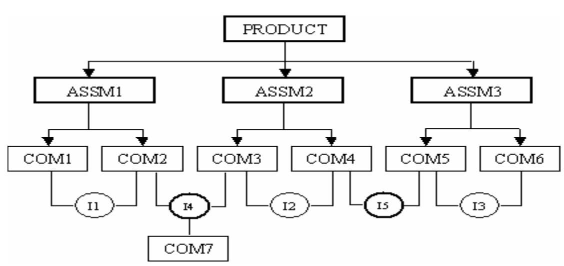

Figure 18. Product structure

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 of that 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.

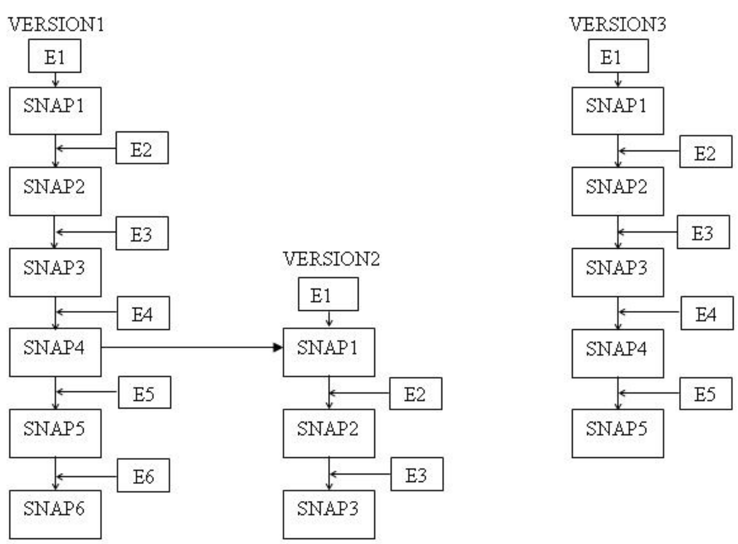

Figure 19. Version-tree with versions, snaps and events

4.3.6 Audio-video clip for an event

All the proceeding 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 the first, preliminary version, we have concentrated on development of the core modules of the framework. The remaining, future work is discussed in Section 7. The overview of the implemented prototype – called IDEA-SUSTAIN - is shown in Figure 20.

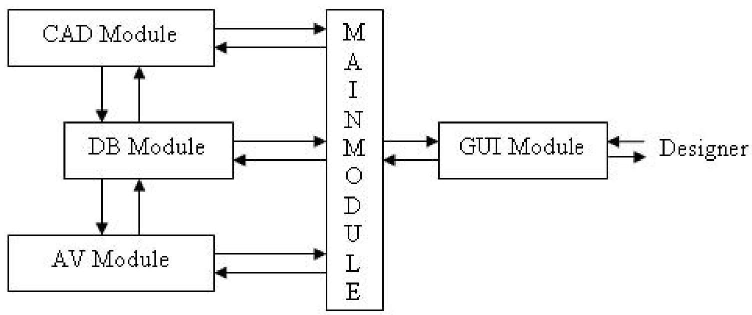

Figure 20 Overview of implemented prototype “IDEA-SUSTAIN”

The prototype consists of the following five modules

- Main Module: It is the program 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 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 program 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 program, and contains 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 program 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 (called Idea-Sustain) 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, and, (3) exploring the events via the video/audio clips attached to the version tree.

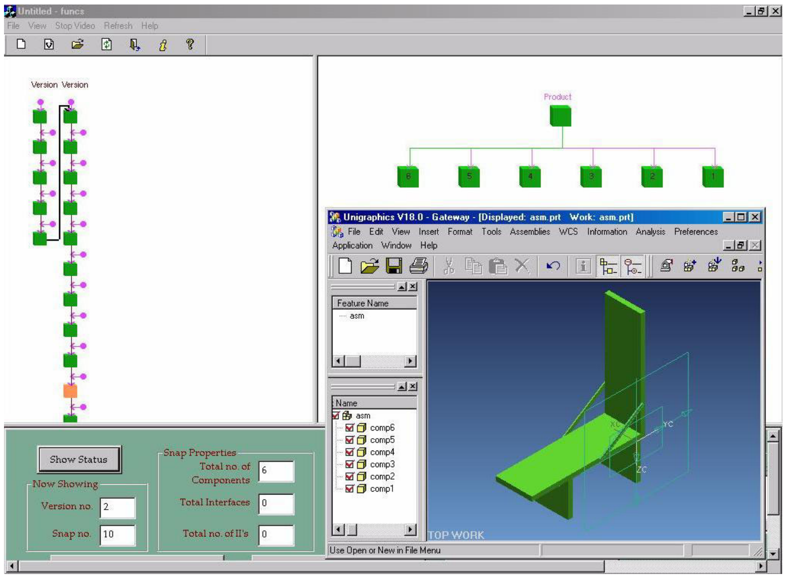

Figure 21: Exploration of Product Structure & Version Tree on the Software

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

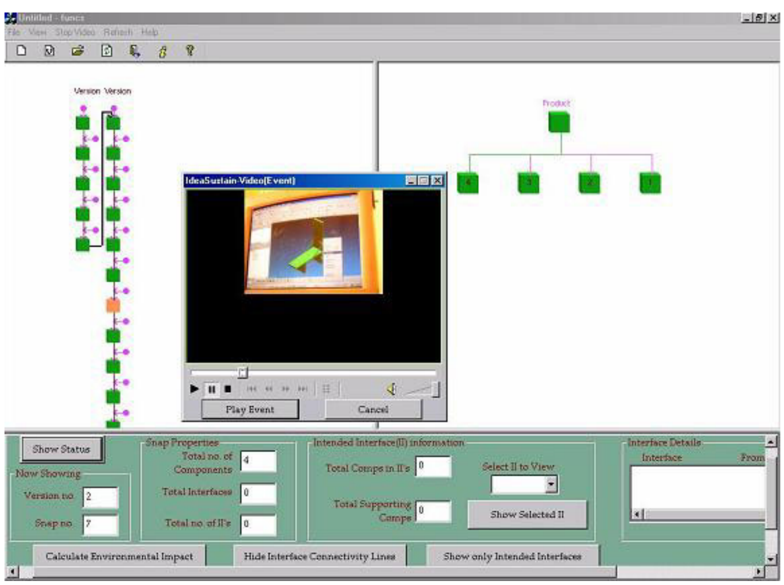

Figure 22: Exploration of an Event Using the Software

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

7. Evaluation, discussion and future work

A user evaluation of the Idea-Sustain software was done in the following way. Two designers were given a brief of one problem each, and asked to solve these problems, first using paper and pen for sketching, and then using the Idea-Sustain software for embodiment. The design process was taken to be complete when designers produced the final drawings and bill of materials. Subsequently, two groups of users (three individual designers in each), who represented re-designers interested in understanding the above design processes, their outcomes and rationale, were asked to first go through the conventional documents produced (problem brief, requirement list, final drawings and bill of materials, along with the sketches and scribbles made by the designer). Followed by this, they were individually interviewed by the corresponding designer to evaluate their depth of understanding.

Depth of understanding of rationale was judged based on the user response to queries about the following ten categories of information:

- What the criteria used in the design process were
- What the alternatives considered were
- How the alternatives were evaluated
- What alternative was decided on and why
- What the critical issues encountered in the design process were
- What the major changes made during design were
- Construction details of the components in the final embodiment
- Location details of components in the final embodiment
- Details of operation of these components and assembly.
- Details of purpose of these components and assembly.

After these evaluations were over, the groups were asked to individually develop an understanding of the design process for the problem they have not focused on in the above phase, this time using the IDEA-SUSTAIN software, problem brief and requirements list, along with the sketches and scribbles made by the designer. At the end of this phase, the individuals in these were interviewed again by the corresponding designers for evaluating their state of understanding of these design processes. These two sets of evaluation were compared to understand the relative influence of the software as a tool for understanding rationale as opposed to conventional documentation. The results are as follows.

The essential conclusions from the above table are:

- Individually for each of the problems (and of course for both the problems taken together), the average level of understanding is better after using Idea-Sustain software than after using conventional documentation. This signifies the influence of IDEA-SUSTAIN in supporting rationale capture and reuse.

- There is significant difference in understanding in case of Problem 1, while the improvement is marginal in case of Problem 2. This indicates the importance/influence of the designer in inputting rationale in the first case, and evaluating people for their understanding of this rationale.

Table 2. Results of Evaluation: User Understanding Levels

Problem1 by designer1 Using conventional documents			Problem2 – by Designer2 Using Idea-Sustain
	Average Score in Eval (1-5)		Average Score in Eval (1-5)
Group1 (1=best 5=worst)	User1	1.8	2.6
	User2	2.1	2.7
	User3	2.1	2.1
	Average	***2.0***	***2.46***
Problem2 – conventional documents			Problem1 – Idea-Sustain
Group2 (1=best 5=worst)	User1	2.0	1.7
	User2	2.0	1.3
	User3	3.5	1.6
	Average	***2.5***	***1.53***
Av. Group1-2		**2.25**	**2.0**

To summarise, IDEA-SUSTAIN framework presented above does the following things.

- It automatically creates a product structure as the 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.
- What we can and cannot do currently are the following. First is, not all snap categories identified in Section 4.3.3 have been implemented – currently only product structure related changes are captured. Second is, early (e.g., sketching/problem understanding) and late processes (usage and after-usage related) are currently not captured. No detailed rationale partitioning is currently implemented, and if there is no designer activity that can be recorded in audio/video clips, little rationale will be there to be captured. Also, the evolution of requirements is currently not captured in an explicit sense.
- Evaluation is currently at the level of whether a real design process can be done and explained using this, and not a comparative evaluation to see how this supports better understanding or redesigning than by conventional designing or rationale capture means.

8. Acknowledgements

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

9. References

[1] S. J. Fenves, R. D. Sriram, R. Sudarsan and F. Wang, "A product information modeling framework for Product Lifecycle Management", Proceedings of PLM Symposium July 16-18, 2003, Bangalore, India.

[2] Ullman, D G, "The Mechanical Design Process", 2nd Edition, McGraw-Hill, 1997.

[3] McGinnis B D, Ullman D G, "The Evolution of Commitments in the Design of a Component", ASME Journal of Mechanical Design, 1991.

[4] Blessing L T M, "A process based approach to computer supported Engineering Design", Thesis, University of Twente, Enschede, The Netherlands, 1994.

[5] Chakrabarti A, Bligh T P, "An approach to functional synthesis of solutions in mechanical conceptual design. Part 1: Introduction and knowledge representation", Research in Engineering Design, No.6, 1994, pp.127-141.

[6] Ullman D G, " A Taxonomy of mechanical Design", ASME International Conference on Design Theory and Methodology, DTM89, Qubec.

[7] Ullman D G, "Design Histories: Archiving the evolution of Products", Proceedings of the DARPA workshop on Manufacturing", February 5-6, 1991, Salt Lake City, Utah.

[8] Jintae Lee, " Extending the Potts and Bruns Model for Recording Design Rationale", Proceedings of 13th International Conference on Software Engineering, May 1991.

[9] Minneman S L, " The social construction of a Technical Reality: Empirical Studies of group engineering design practice", Thesis, Dept. Mechanical Engineering, Stanford University, 1991.

[10] Kuffner Tom A, Ullman D G, "The information requests of mechanical design engineers", Design Studies, Vol.12, No.1, 1991, pp.42-50.

[11] Tang John C, "Listing, Drawing and Gesturing in Design: A study of the use of shared workspaces by design teams", Thesis, Dept. Mechanical Engineering, Stanford University, 1989.

Corresponding author:
Amaresh Chakrabarti,
Innovation Design Study and Sustainability Laboratory (IdeasLab)
Centre for Product Design and Manufacturing,
Indian Institute of Science, Bangalore – 560012.India.
Tel: 0091-80-22932922
Fax: 0091-80-23601975
E-mail: ac123@cpdm.iisc.ernet.in
URL: http://cpdm.iisc.ernet.in/ac.htm