

Automated Capture of Design Knowledge Using a Virtual Creature Design Environment

R.C.W. Sung¹, J.M. Ritchie¹, T. Lim¹, H.J. Rea², J.R. Corney³

¹Department of Mechanical Engineering, Heriot-Watt University, Edinburgh, EH14 4AS, United Kingdom. E-mail: r.c.w.sung@hw.ac.uk

²The University of Edinburgh, Darwin Building, The King's Buildings, Edinburgh, EH9 3JR, United Kingdom

³Design, Manufacture & Engineering Management, University of Strathclyde, Glasgow, G1 1XJ, United Kingdom

Keywords: Knowledge Capture, Computer Aided Design (CAD), User Logging, Knowledge Management, Design Rationale

Abstract

Current-generation computer-aided design (CAD) tools are excellent for designing and recording the eventual solution to an engineering problem but they are not ideal at recording the design knowledge which is generated by the engineer, during the session. Being able to view this information would be extremely beneficial for engineers who need to look at past designs to find out why design decisions were made. Engineers could manually record all their decisions but this would be time-consuming and disruptive to the creative process. The solution proposed in this paper involves automating the user-logging process and performing automated analysis of the generated log files to extract the design knowledge.

A virtual creature design game environment called BAMZOOKi is used to demonstrate how design knowledge can be extracted and represented and then used as a basis for an interactive online help system.

1 Introduction

An engineering drawing or 3D CAD model only defines one instantiation of a solution to a design process. Although many industries (e.g. aerospace and defence) have established systems for documenting the analysis done to verify the functionality of a given assembly or component, much of the knowledge developed and used during a design process is lost. Although the solution to a problem may be recorded, the processes by which this solution was derived and the constraints discovered and defined during the process go largely un-archived. It may be possible to ask the engineer to manually log all the design knowledge generated during a design session, but *“it takes designers’ time away from design, it interrupts designers’ normal thinking process, and it is often impossible for designers to articulate their tacit knowledge explicitly”* (Jin & Ishino, 2006). Issues like this may dissuade engineers from recording their design knowledge, but this would be unfortunate, since a concise and accessible summary of an product’s design process would aid engineers who, sometimes years later, have to revisit product designs to upgrade or modify them. Even though some current commercial CAD and CAE systems have the facility to log the actions of users, they do not attempt to extract the design rationale that is embedded in the data.

Motivated by these needs the authors are undertaking research into the automated capture and re-use of design, design activity and design process data associated with an individual engineer’s behaviour. A key issue within the project was that industrial partners involved in the research were reluctant to allow user logging in-situ on their secure design systems for security reasons. Therefore, an alternative

CAD design environment had to be found and implemented. To allow progress towards the goal of automated design rationale capture, the authors looked for a ring-fenced non-trivial design environment that had an intuitive user interface and integrated design verification tools.

In considering these criteria, the CBBC BAMZOOKi™ design application was identified as providing a self-contained ‘design-build-test’ CAD environment which mimicked an analytical try-and-test cycle commonly used within industry; the structure, operation and interface of which made it an ideal platform for this research. The BAMZOOKi application allows users to construct virtual mechanical creatures (Zooks) from building block elements and modify the properties of these elements to change either the Zook’s form or motion and, subsequently, performance. The environment also includes a test section where the performance of the Zook can be measured against a set of standard tasks. The version of the BAMZOOKi software distributed by CBBC does not support any continuous user logging, but a bespoke version was created by the game developer for use in this research.

The rest of this paper is structured as follows: Section 2 presents a brief literature review of knowledge capture research and Section 3 describes the BAMZOOKi product design environment. Section 4 outlines the proposed experimental methodology. The experimental procedure is defined in Section 5 and results are given in Section 6. Finally, conclusions are drawn in Section 7.

2 Literature Review

The logging of design processes is an important problem because the design knowledge extracted from it is not only useful in the training of new users, but it also means that vital knowledge is not lost from a company when an engineer retires or moves to another company (Matsumoto *et al*, 2005) particularly for long-life products (Ding *et al*, 2007). It is also stated in Conway *et al* (2008) that: “Current documentary approaches are not sufficient to capture activities and decisions in their entirety and can lead to organizations revisiting and in some cases reworking design decisions in order to understand previous design episodes.” To overcome this problem, automated and unobtrusive logging of the engineer could be utilised by adding direct user-logging functionality to the design environment and post-processing the data into a readable and usable format. Campbell *et al* (2005) describes a method for recording design activity on a PC as well as information “push”, such as providing links to relevant documents as a user undertakes a task. In a similar approach, Stump *et al* (2005) describe a system which monitors user behaviour, and interactions with Microsoft Windows are logged.

Jin and Ishino (2006) use the combination of user activity logging and data mining to automatically extract design activity knowledge that is embedded in a 2D CAD design session for the design of automobile doors.

Shipman and McCall (1997) present two automatic design rationale systems called Hyper-Object Substrate (HOS) and PHIDIAS, and the two systems allow the capture of various media related to CAD and network design. However, they neither automatically represent the captured design knowledge and information in a formalised or understandable manner nor push this at the user.

Iver *et al* (2006) presents a system for extracting design intent from legacy CAD drawings by automatically extracting the geometrical and textual entities from the drawing. However, no final results are yet available.

Brandt *et al* (2008) presents a system called ‘Process Data Warehouse’ (PDW) that allows the capture and re-use of design knowledge during a design of a chemical reactor. Currently, the capturing of work processes and decisions is not supported and the current system can only handle a small amount of instance data.

A virtual environment, called Virtual Training Studio (VTS), is presented by Schwartz *et al* (2007), where users are trained to perform assembly tasks and assistance is offered to them, in the form of video clips and 3D animations, if requested or if errors are made. In relation to the work in this paper, VTS concentrates on the assembly stage but not the design stage of the product development cycle.

Similarly, work at by Robinson *et al* (2007) has demonstrated the use of immersive virtual environments to derive design process knowledge from logs of user activity. Much of this work has been based on cable harness design and has provided an ideal example for computer aided design (CAD) as described in this work.

3 CAD Experimental Environment

The CAD experimental environment chosen was the BAMZOOKi package which was developed for a CBBC TV children's programme, by Gameware Development Ltd, and comprised of a toolkit which is freely available for download and standalone use. This enabled a constrained design task to be defined as well as providing the potential for data from a very large number of users (or designers); something simply not possible from the industrial partners within the research. Within BAMZOOKi, Zooks are virtual creatures that users create to perform with specific mechanical behaviours. Users can then compete in a number of ways with other players' creations to compare their design's performance. A successful design depends on players developing a complex understanding of how to create and design a Zook, and trials in the design environment include sprinting, pushing and hurdles.

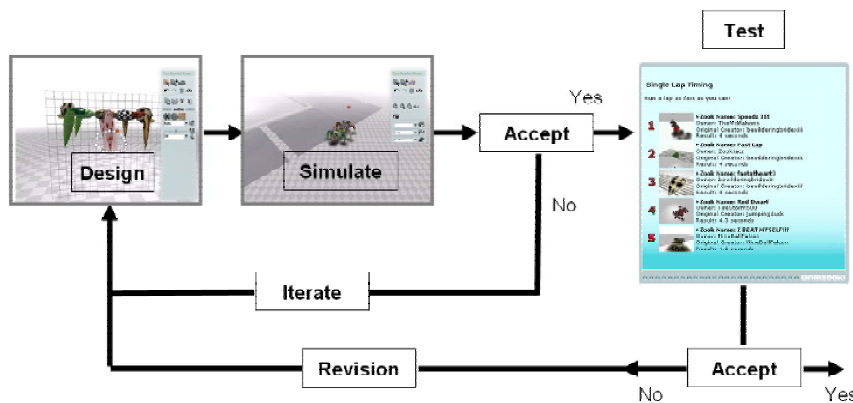


Figure 1 - BAMZOOKi Design Process

The BAMZOOKi Zook-kit is an intuitive 2D design environment and its design process, illustrated in Figure 1, closely mimics a typical engineering design process, which involves creating a design, testing it and, depending on the test results, making further design revisions. Figure 2 shows the user interface that allows users to create their Zook, make revisions to it and then test how well it performs.

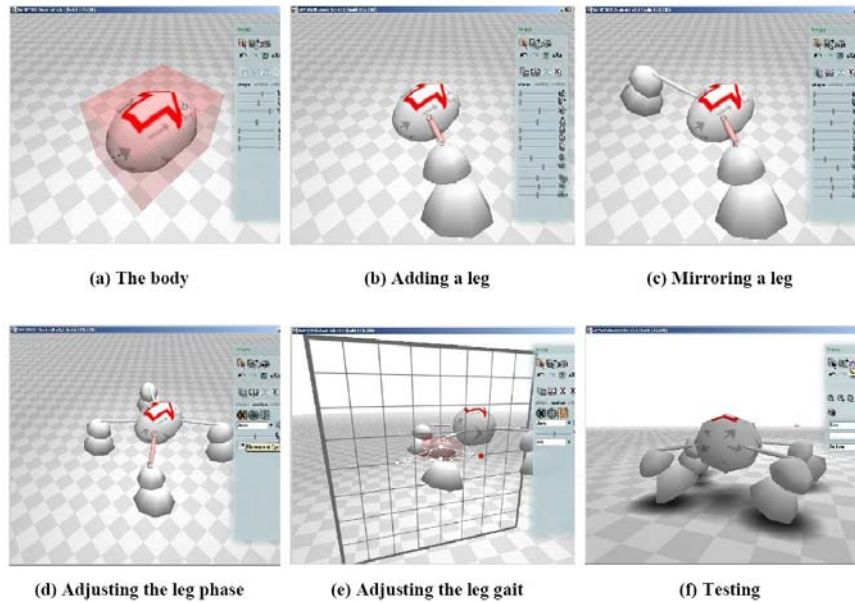


Figure 2 - Creating a Zook

4 The Experiment

4.1 Aims

The objective of logging the design evaluation sessions was to investigate the processes by which people “discover” what aspects of their Zook can best be changed to meet a *specific* design performance goal. There are many options open to the designer to improve the Zook’s performance: weight reduction; leg lengthening; change of gait, etc. As they learn about the Zook’s capability, each designer “discovers” which of these parameters has the most effect through a trial and error approach. The research challenge was to automatically extract the “knowledge” created during the design process from the log files and show how this could be represented and pushed at the user in an understandable format; demonstrating the potential capabilities and functionality that should be embedded within in proprietary CAD systems.

4.2 . Methodology

Each participant in the experiment was given a default Zook (known as the “Baseline”) with a performance goal they were required to meet, e.g. achieve a sprint speed of 40cm/s. The participants then proceeded to modify the design to meet this specification.

The objective of the logging capability was to unobtrusively capture a complete record of each product design session without interrupting the user’s cognitive design thought process. The log data is stored in XML (Extensible Markup Language), which is a W3C-recommended general-purpose markup (Extensible Markup Language 2008) that facilitates the sharing of data across different information systems; also, XML languages are easy to design and process.

5 Experimental Procedure

5.1 Baseline Zook

The design session starts with a baseline Zook, as illustrated in Figure 3, which has a ‘below-average’ performance in the three test tasks (sprint, hurdles and block push), and its design deliberately incorporates these ‘poor’ elements so that they can be identified and modified by the designer to improve performance.

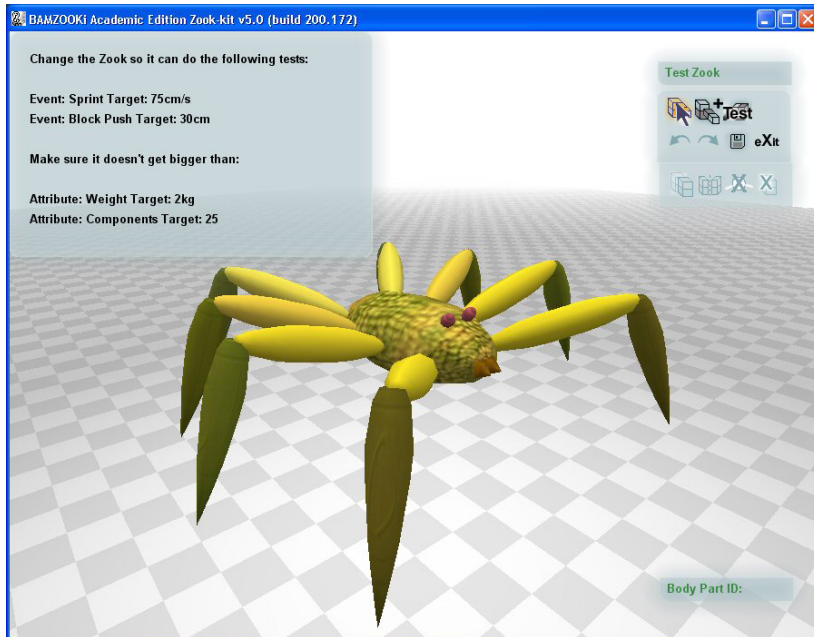


Figure 3 - Baseline Product

Several variations of the baseline Zook may be used to provide different levels or complexities of the starting ‘problem’. Example design problems and possible solutions are detailed in Table 1

Problem	Potential Solution
Sprint:	
The zook walks rather than runs. The zook takes small steps. Legs are too close to increase stride length. Legs are too short to have a big stride length. Zook appears to ‘limp’ or is ‘jerky’.	Increase the gate cycle speed. Increase the stride length. Increase body length and increase leg spacing. Increase leg length. Adjust phase of legs within the gait cycle
Block Push:	
Zook is too small to push the blocks. Zook legs contact the blocks rather than the body / head. The head makes contact but zook skids on the blocks.	Make the zook bigger. Make head larger or change angle of front legs, (etc). The front of the zook has a point contact, make the jaws larger and wider to produce a more stable contact.
Hurdles:	
Zook gets stuck on the larger hurdles. Zook cannot get onto the larger hurdles. The zook is unstable and sometimes falls over.	Increase the zook’s ground clearance. Change gait cycle to increase step height. Increase width of the zook.

Table 1 - Possible Zook Problems and Solutions

5.2 Goal

The goal of each designer was to make modifications to the product which will lead to it achieving a higher ‘Gold’ level of performance. This ‘Gold’ performance must be set at a realistic level (average) that can be achieved by a typical user within the timescale of a test session. This is unlikely to be a best-in-class performance and further improvements may be expected from a longer session or with more experienced users. Three levels of users were anticipated; beginners, intermediate users and experts; therefore, three sets of goals were developed: apprentice, master and wizard respectively. The performance parameters for these levels are shown in Table 2.

Apprentice Level			
Physical (approx. values):		Performance:	
Height:	no limit	Sprint:	75 cm/s
Width:	no limit	Block Push:	30 cm
Length:	no limit		
Weight:	2 kg		
Components:	25		
Master Level			
Physical (approx. values):		Performance:	
Height:	no limit	Sprint:	75 cm/s
Width:	no limit	Block Push:	30 cm
Length:	no limit	Hurdles:	30 cm/s
Weight:	2 kg		
Components:	25		
Wizard Level			
Physical (approx. values):		Performance:	
Height:	15 cm	Sprint:	100
Width:	40 cm	cm/s	
Length:	35 cm	Block Push:	50 cm
Weight:	2 kg	Hurdles:	75 cm/s
Components:	25		

Table 2 - 'Gold' Performance Parameters

6 Results

The design software was exhibited and used by attendees at two major science festivals which allowed nearly 800 log files to be obtained. The majority of design sessions were Apprentice Level at both venues. A statistical analysis of the log files was performed and the results are presented in a prior paper (Rea *et al* 2007).

6.1 Alternative Design Process Representations

To allow the design knowledge in the log files to be stored in repositories such as Product Life Management (PLM) systems, more structured and formal representations of the knowledge were required. Therefore, various representations were chosen, which were then automatically generated by parsing of the log files.

6.1.1 XML and PSL Representation

A log file monitor application, written using C#, was used to analyze the log file in real-time and each time the user made a design change, an XML representation of the event was output to a file. Whereas the log file contained a record of all user activities, this secondary output file only contained activities that had resulted in a physical change of the product. After a few design changes, an XML file, such as the one shown in Figure 4, was obtained.

```
<?xml version="1.0" encoding="UTF-8" ?>
<bamzooki event>
  <change_leg_size>13</change_leg_size>
  <change_body_size>1</change_body_size>
  <change_ik_point>3</change_ik_point>
  <change_ik_point>2</change_ik_point>
  <change_ik_point>4</change_ik_point>
</bamzooki_event>
```

Figure 4 - XML File Fragment

Having this structured representation allows a database to be created that contains these XML fragments. Then, by monitoring the user's XML output in **real-time** during a design session, any XML output sequences that match those in the database will allow the system to **predict** what activity the user is about to carry out. Once this pattern is detected, any assistance that is relevant to the detected activity can then be 'pushed' at the user. A prototype information push system based on this concept is presented in section 6.2.

In addition to the XML statements, PSL (Process Specification Language) statements were also automatically generated. PSL is an ISO standard that was developed to model the processes that occurred during manufacturing activities, and since the format is in a machine-readable form, it is a widely-supported format (Gruninger and Bock 2005). Just like XML, the PSL representation is automatically generated in real-time during a design session. As illustrated in Figure 5, XML and PSL representations are currently only generated for three main activities, i.e. changing leg length, body size and stride pattern. However, future work will involve automatically generating XML and PSL representations for all known possible activities.

	Change Leg Length	Change Stride Pattern	Change Body Size
XML	<?xml version="1.0" ?> <bamzooki_events> <change_leg_length>1</change_leg_length> </bamzooki_events>	<?xml version="1.0" ?> <bamzooki_events> <change_stride_pattern>1</change_stride_pattern> </bamzooki_events>	<?xml version="1.0" ?> <bamzooki_events> <change_body_size>1</change_body_size> </bamzooki_events>
PSL	(activity ChangeLegLength) (subactivity DragLeg ChangeLegLength) (subactivity DragMenuSlider ChangeLegLength)	(activity ChangeStridePattern) (subactivity DragIKPoint ChangeStridePattern)	(activity ChangeBodySize) (subactivity DragBody ChangeBodySize) (subactivity DragMenuSlider ChangeBodySize)
Plain-English Syntax	Change left front leg to the following dimensions (WxHxD): 2.14 2.27 0.62	Move IK Point 2 on right front leg to vector: 0 0.5 2.96	Change body to the following dimensions (WxHxD): 28.4 24.3 15.8

Figure 5 - XML and PSL Representations

6.1.2 IDEF0 Diagrams

In order to produce a readable flow process diagram of each design session IDEF0 diagrams were used. IDEF0 (Integrated computer-aided manufacturing DEFinitions) diagrams were invented to provide a method to visualize and describe the design processes that occur during a design task (Integrated Definition Methods, 2009). As shown in Figure 6, an IDEF0 diagram consists of the main FUNCTION component which represents the main activity taking place. The INPUT component is the data or object that is transformed by the FUNCTION into the OUTPUT. Finally, the CONTROL component specifies the constraints on the system and the MECHANISM is the means by which the activity can take place.

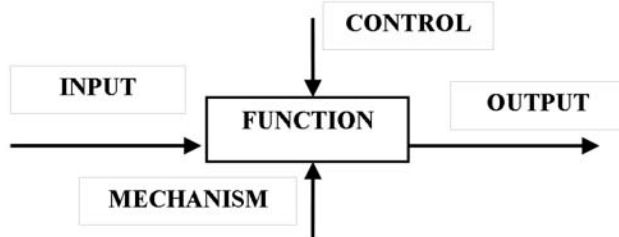


Figure 6 - IDEF0 Diagram

By parsing the XML log files, IDEF0 diagrams can be automatically generated using a spreadsheet macro, as shown in Figure 7. This figure shows part of the IDEF0 diagram that was generated for one particular log file, but the same process can be easily repeated for all the other log files that were collected during the experimental trials.

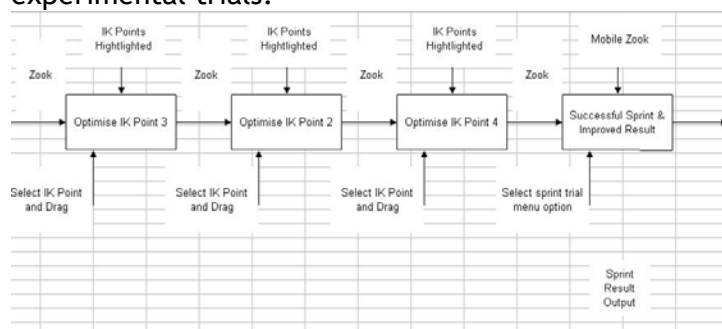


Figure 7 - A Typical BAMZOOKi Product Design Process IDEF0 Diagram

By visual inspection of the generated IDEF0 diagram, the precise steps that were taken to create the final design can be easily seen and understood; this information can be used to train new users on the best practices required for a good design or, in a real industrial environment, be stored for future use and reference in a PLM system. Crucially, this is the first time that design process maps have been automatically generated for a CAD-like environment.

6.1.3 English-Syntax Instructions

Further assistance could be offered to new users in the form of clear and concise instructions on how to create a successful design. By automated parsing of the log file and identifying patterns of activity from the PSL output, a set of English-syntax instructions can be generated which can then be used as an aid for new users or, again, stored in a PLM system to describe the change. An example of such automatically generated instructions is shown in Figure 8, which also contains the XML and PSL equivalents.

XML	PSL/English-Syntax Instruction
<pre><event desc = "Zook-Physical" > <time hour = "09" minutes = "22" seconds = "44" seconds_milli = "108" > </time> <ui_interaction > <parameter_change Height = "1.464130401611328" body_part = "1" > </parameter_change> </ui_interaction> </event></pre>	<p>(subactivity Change-BodySize_1 ImproveZookPerformance)</p> <p>Change main body to the following height: 1.46</p>
<pre><event desc = "Zook-Physical" > <time hour = "09" minutes = "22" seconds = "49" seconds_milli = "608" > </time> <ui_interaction > <parameter_change Width = "0.6891304850578308" body_part = "4" > </parameter_change> </ui_interaction> </event></pre>	<p>(subactivity Change-LegSize_4 ImproveZookPerformance)</p> <p>Change right back leg to the following width: 0.69</p>
<pre><event desc = "Zook-Dynamic" > <time hour = "09" minutes = "22" seconds = "57" seconds_milli = "61" > </time> <drag_interaction > <parameter_change body_part = "6" ik_point = "3" position = "vec -0.000000 0.986596 0.008840" > </parameter_change> </drag_interaction> </event></pre>	<p>(subactivity Change-IKPoint3 ImproveZookPerformance)</p> <p>Move top centre IK point on right back leg 2 unit(s) up from origin</p>

Figure 8 - A Plain-English Syntax Description of the Design Process

By having the user view the list of English-syntax instructions and performing these changes one at a time, this will allow them to become quickly familiarized with how to create a good design. As more experience is gained by the user, more efficient design methods may be discovered which can then be used to update the information push system. Again, this points to the potential benefits that would be achieved if real CAD systems were to have this type of user-logging capability incorporated within them, whilst being used in actual engineering environments.

6.1.4 DRed

Another available design rationale logging tool is called DRed (Design Rationale Editor). This was created to allow design rationale generated during a meeting to be manually recorded in real-time (Kim *et al* 2004). A DRed diagram consists of individual colour-coded symbols that are linked by arrows which signify the workflow direction. The colours represent whether a design decision has resulted in a successful solution and the symbols represent *answers*, *issues*, *pro* and *con* arguments and text statements. The research in this paper has managed to automatically generate DRed diagrams by parsing the log files using a spreadsheet macro. Figure 9 shows a DRed diagram of a design session and the actions that were performed by a logged user.

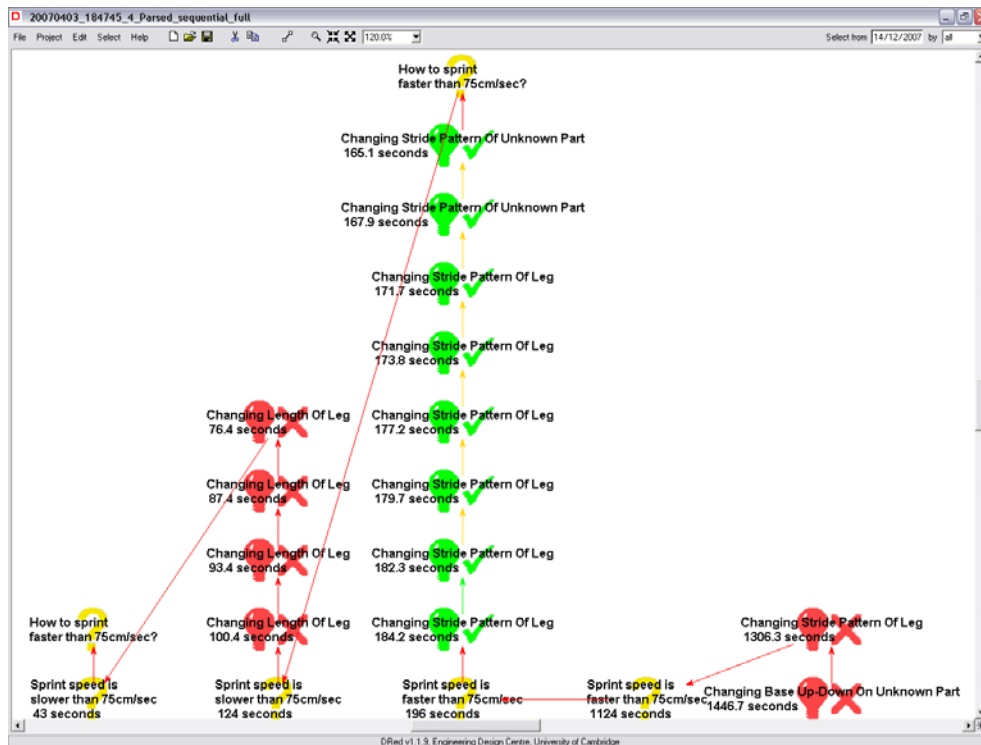


Figure 9 - DRed Representation of BAMZOOKi Design Session

In this figure, all the yellow question mark elements represent an issue and the outcome of a sprint trial, the red light bulbs with a cross next to it represents design changes made by the user which have not resulted in improving the sprint speed of the product, and the green light bulbs with a tick next to them represent design changes that has resulted in improving the performance of the product. Furthermore, each of the arrows that link the elements is colour-coded to represent the time interval between the user actions. If the action took less than 5 seconds to complete, the arrow is green, if the action took between 5 and 10 seconds, the arrow is amber, and if the action takes more than 10 seconds, the arrow is red. From the coloured-arrows, it is possible to see which actions are taking too long to complete by the designer and helps indicate what aspects the designer was having difficulty with. Note that not only is the successful solution captured in this example - as shown in green - but other failed or less successful solutions are also represented automatically overcoming one of the classic problems with design rationale capture mentioned previously.

6.1.5 Annotated Video Clips

As a means of representing a properly described design session which could be files for future use or as a training aid to help engineers create a successful design, video captures of an expert-user creating a design were taken and then automatically subtitled to explicitly describe the user's actions; these are automatically embedded into the video clip after the session is completed but future work will perform this in real time. During the playback, the subtitles appear at the same time the corresponding user action takes place, as shown in Figure 10.



Figure 10 - Annotated Video Capture of a Design Session

To generate an annotated video capture, the first step involved using a hardware video capture device to record the output from the computer graphics card, while an expert-user creates a design. Next, the log file is parsed which generates an output text file, known as a subtitle script, that contains the subtitles and the time at which each one is to be displayed in the video clip. The final step then involves importing both the original video capture and the subtitle script into the Adobe Encore DVD authoring application and combining both items to generate the final annotated video clip. This process can be performed for any design session carried out during the experimental phase of this work.

6.1.6 Storyboard Representation

By combining all the various design knowledge representations into one 'storyboard' format, a user who views this can ascertain all the key stages that have happened, during a design session, and understand the consequences of making each design decision (Figure 11).

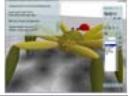

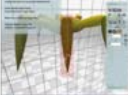



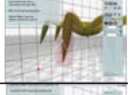



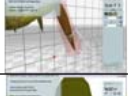


Time (secs)	DRed	PSL	English Syntax	Screengrab
		(activity PerformSprint)	Select sprint trial in drop-down menu	
14	How to sprint faster than 75cm/sec?	(subactivity FailedSprint PerformSprint)	Sprint result of 44.8cm/sec has not achieved target	
43	Changing Stride Pattern Of Leg 43.3 seconds	(activity ImproveZookPerformance) (subactivity ChangeIKPoint3 ImproveZookPerformance)	Move top centre IK point on right second-back leg 2 unit(s) up from origin	
45	Changing Stride Pattern Of Leg 45 seconds	(subactivity ChangeIKPoint4 ImproveZookPerformance)	Move bottom left IK point on right second-back leg 2 unit(s) left from origin	
46	Changing Stride Pattern Of Leg 46.5 seconds	(subactivity ChangeIKPoint2 ImproveZookPerformance)	Move bottom right IK point on right second-back leg 2 unit(s) right from origin	
49	Changing Stride Pattern Of Leg 49.1 seconds	(subactivity ChangeIKPoint3 ImproveZookPerformance)	Move top centre IK point on right back leg 2 unit(s) up from origin	
51	Changing Stride Pattern Of Leg 50.7 seconds	(subactivity ChangeIKPoint2 ImproveZookPerformance)	Move bottom right IK point on right back leg 2 unit(s) left from origin	
53	Changing Stride Pattern Of Leg 52.6 seconds	(subactivity ChangeIKPoint4 ImproveZookPerformance)	Move bottom left IK point on right back leg 2 unit(s) right from origin	
60	Changing Stride Pattern Of Leg 60.2 seconds	(subactivity ChangeIKPoint3 ImproveZookPerformance)	Move top centre IK point on right front leg 2 unit(s) up from origin	
62	Changing Stride Pattern Of Leg 61.8 seconds	(subactivity ChangeIKPoint2 ImproveZookPerformance)	Move bottom right IK point on right front leg 2 unit(s) right from origin	
63	Changing Stride Pattern Of Leg 63.2 seconds	(subactivity ChangeIKPoint4 ImproveZookPerformance)	Move bottom left IK point on right front leg 2 unit(s) left from origin	
	How to sprint faster than 75cm/sec?	(activity PerformSprint)	Select sprint trial in drop-down menu	
75	Sprint speed is slower than 75cm/sec 14 seconds	(subactivity FailedSprint PerformSprint)	Sprint result of 79.1cm/sec has achieved target	

Figure 11 - Storyboard Representation of a Design Session

6.2 Information Push System

After extracting the design knowledge required to achieve a successful sprint, block push or hurdles trial result from expert designers, this information pattern can then be stored within a PSL format and then be ‘pushed’ to inexperienced users who are working on the same design task.

By using the same C# log file monitor application mentioned in section 6.1.1, it is possible to search for occurrences of specific events. For example, if a user were to consecutively fail a sprint trial more than once, the log file monitor can identify this.

```

- <event>
  <time hour="15" minutes="25" seconds="04" seconds_milli="439" />
  <achievement comment="cm/sec" name="Trial: Sprint" score="43.2" />
</event>
+ <event desc="UI Only">
+ <event desc="UI Only">
+ <key_event zook_version="002">
- <event desc="Zook-Physical">
  <time hour="15" minutes="25" seconds="22" seconds_milli="876" />
- <ui_interaction>
  <parameter_change Width="2.252173662185669" body_part="1" />
</ui_interaction>
</event>
+ <event desc="UI Only">
+ <event desc="UI Only">
- <event desc="UI Only">
  <time hour="15" minutes="25" seconds="29" seconds_milli="752" />
- <ui_interaction>
  <test_trial name="Sprint" />
</ui_interaction>
</event>
- <event>
  <time hour="15" minutes="25" seconds="37" seconds_milli="580" />
  <achievement comment="cm/sec" name="Trial: Sprint" score="41.9" />
</event>

```

Figure 12 - Log File Showing Two Failed Sprint Trials

In the log file fragment shown in Figure 12, the user has started the design session by performing a sprint trial which results in a sprint speed of 43.2cm/sec. Since this value is less than the target value of 75cm/sec, this trial result is deemed a failure. Next, the user increases the width of the Zook body and then performs another sprint trial. Once again, a failed sprint result was achieved, with a value of 41.9cm/sec, so the log file monitor identifies this and issues a command to BAMZOOKi to display a dialogue box to the user to ask if any help is required, as shown in Figure 13.

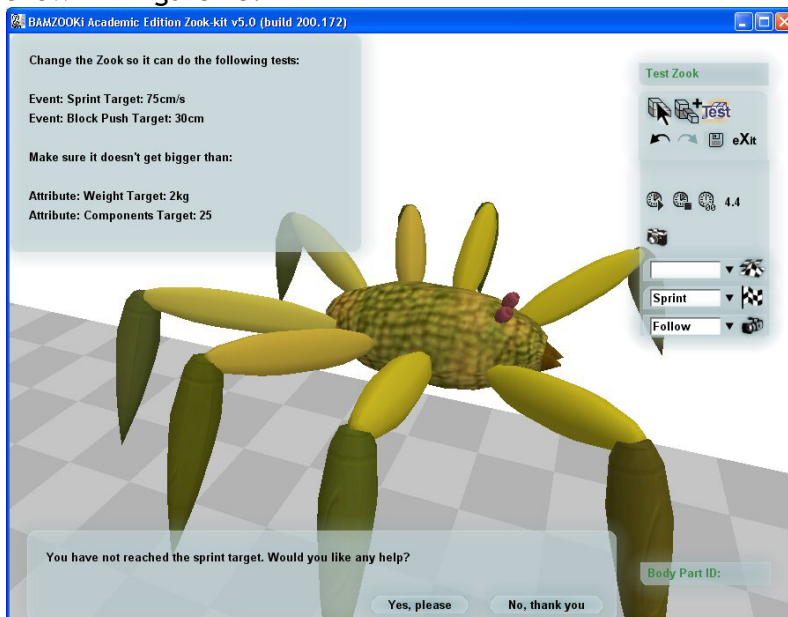


Figure 13 - Sprint Online Help

The user can then select to accept or reject the help by pressing the corresponding button in the dialogue box. If help is accepted, a page will then be displayed which explains to the user the design changes that are required to achieve a successful trial result. The information on this help page consists of the design knowledge that was acquired in the log file analysis mentioned previously. This information push system demonstrates three instantiations at present within the system. This is the first example of effective, automated real-time information push in a CAD environment facilitated and driven only through user logging and behavioural analysis between the user and the system.

7 Conclusions

This paper has presented an automated knowledge capture system that unobtrusively logs the designer, whilst they are designing a virtual mechanical creature in a CAD environment. From the analysis of the log files, embedded design knowledge has been successfully extracted and represented using various visual and codified representations. This data was then used to create training tools and a prototype interactive online help system.

Future work will involve continued analysis of the log files to extract more design knowledge and streamlining of the information push system. Furthermore, these tools will be attempted to be integrated within an industrial-based CAD system to allow more realistic design tasks to be logged and analysed.

Acknowledgments

The work presented herein was undertaken under the aegis of the Knowledge and Information Management (KIM) Through-Life Grand Challenge Project (www.kimproject.org) funded primarily by the Engineering and Physical Research Council (EPSRC - Grant No EP/C534220/1), the Economic and Social Research Council (ESRC - Grant No RES-331-27-0006) and Heriot-Watt University's Innovative Design and Manufacturing Research Centre (IMRC- Grant No GR/S12395/01). We would also like to thank CBBC for giving us permission to use BAMZOOKi for our research, and Dylan Banarse of Gameware Development Ltd (<http://www.gamewareddevelopment.co.uk/>) for his assistance in creating the specifications for the bespoke BAMZOOKi program.

References

- Brandt, S.C., Morbach, J., Miatidis, M., Theissen, M., Jarke, M., Marquardt, W. (2008). An ontology-based approach to knowledge management in design processes, *Computers & Chemical Engineering*, Vol. 32, 320-342.
- Campbell, D.R., Culley, S.J., McMahon, C.A., and Coleman P. (2005). A Methodology for Profiling Computer Based Design Activities", *International Conference on Engineering Design (ICED 05)*.
- Conway, A.P., Giess, M.D., Lynn, A., Ding, L., Goh, Y.M., McMahon, C.A., Ion, W.J. (2008). Holistic Engineering Design: A Combined Synchronous and Asynchronous Approach, *ASME IDETC/CIE 2008*, Aug 3-6, New York, USA, DETC2008-49340.
- Ding, L., Ball, A., Matthews, J., McMahon, C., Patel, M. (2007). Product representation in lightweight formats for product lifecycle management. *Proceedings of the 4th International Conference on Digital Enterprise Technology*, Bath UK, September 19-21, 2007.
- Extensible Markup Language: <http://www.w3.org/xml/> (Last viewed on Dec 11th 2008)
- Gruninger, M., Bock, C. (2005). PSL: A semantic domain for flow models, *Software Systems Modeling*, Vol. 4, 209–231.
- Integrated Definition Methods. (2009). <http://www.idef.com/> (Last viewed on Mar. 20th 2009)
- Iyer, G.R., Mills, J.J., Barber, S., Devarajan, V., Maitra, S. (2006). Using a context-based inference approach to capture design intent from legacy CAD. *Computer-Aided Design & Applications*, Vol. 3 (1-4), 269-278.
- Jin, Y., and Ishino, Y. (2006). Daka: Design Activity Knowledge Acquisition through Data-Mining, *International Journal of Production Research*, Vol. 44, No. 14, 2813-2837.
- Kim, S., Bracewell, R.H., Wallace, K.M. (2004). From discourse analysis to answering design questions. *International Workshop on the Application of*

- Language and Semantic Technologies to Support Knowledge Management Processes*, October, Whittlebury, UK.
- Matsumoto, I.T., Stapleton, J., Glass, J., Thorpe, T. (2005). A knowledge capture report for multidisciplinary design environments, *Journal of Knowledge Management*, Vol. 9, No. 3, 83-92.
- Rea, H.J., Corney, J.R., Ritchie, J.M., Sung, R.C.W., Salamon, C. (2007). Automating Digital Capture of Engineering Knowledge, *4th International CIRP-Sponsored Conference (DET2007)*, 19-21 September 2007, Bath, UK.
- Robinson, G., Ritchie, J.M., Day, P.N., and Dewar, R.G. (2007). System Design and User Evaluation of Co-Star: An Immersive Stereoscopic System for Cable Harness Design. *Computer-Aided Design*, Vol 39, 245-257.
- Schwartz M, Gupta SK, Anand DK, Kavetsky R. (2007). Virtual Mentor: a step towards proactive user monitoring and assistance during virtual environment-based training. *Performance Metrics for Intelligent Systems (PerMIS) Workshop*, Gaithersburg, MD, August 2007.
- Shipman III, F.M., McCall, R.J. (1997). Integrating different perspectives on design rationale: Supporting the emergence of design rationale from design communication. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 11, no. 2, 141-154.
- Stumpf, S., Bao, X., Dragunov, A., Dietterich, T.G., Herlocker, J., Johnsrude, K., Li, L., and Shen, J. (2005). Predicting User Tasks: I Know What You're Doing! *Workshop on Human Comprehensible Machine Learning at the Twentieth National Conference on Artificial Intelligence (AAAI-05)*.