INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

DRED 2.0: A METHOD AND TOOL FOR CAPTURE AND COMMUNICATION OF DESIGN KNOWLEDGE DELIBERATED IN THE CREATION OF TECHNICAL PRODUCTS

Rob Bracewell[1], Marina Gourtovaia[1], Michael Moss[2], David Knott[2], Ken Wallace[1], P John Clarkson[1]
(1) Cambridge University, UK (2) Rolls-Royce plc, UK

ABSTRACT
This paper addresses the general issue of software tool support for designers, helping them to structure, to communicate and to document activities of generation, evaluation and decision. Here the focus is on detailed consideration of desired and undesired behavioural relationships among elements of complex design artefacts and with end users. This is an area that has recently been under discussion by proponents and critics of Affordance Based Design methods. Our solution approach is to extend an existing graph based software tool for design rationale capture that has been in widespread use in an international aerospace company for several years. We are integrating its Issue Based Information System (IBIS) based design argumentation with hierarchical Functional Analysis Diagrams (FAD), a form of Concept Map. The resulting software is being tested by practical application on pilot projects in the company, and initial experiences have been very favourable. The new graph element types, bi-directional relationship types between graphs, and supporting navigational facilities are described. Their use is illustrated by example of an integrated hierarchical FAD and assembly geometry model of a gas turbine engine.

Keywords: IBIS, Concept Map, Affordance, Functional Analysis Diagram, TRIZ

1 INTRODUCTION

It has recently been proposed by Meier and Fadel, that the shortcomings of commonly accepted function-based theories of design might be overcome by the adoption of a *relational* theory, which they call Affordance Based Design (ABD) [1]. The core of their proposal is the representation of the complex web of useful and harmful relationships between artefacts, the designers of these artefacts, and their users. They argue that consideration of this web of relationships might give a unified basis for design generation and decision making - “a comprehensive high-level approach to design” [2]. In the field of perceptual psychology Gibson coined the term *affordance* [3] for similar relations describing the perception by an animal of objects in its environment. Meier and Fadel thus adopted the affordance concept as the basis of their design theory and method. They state that alternative relational concepts could be proposed for this task, and that their choice of affordance was primarily based on a claimed theoretical grounding that it confers.

The idea of a design method based on the capture and deliberation of relationships between designers, artefacts and users, significantly predates the proposal of ABD. In the early 1990s, the n-dim Group at Carnegie Mellon EDRC investigated collaborative and participatory design methods, supported by a sophisticated computational environment for both the research, and the delivery of resulting tools to end users [4]. One of the key concepts of the n-dim environment was that of a *flat space of objects* allowing any relevant relationship between any pair of objects to be captured no matter where they reside in whatever hierarchical decompositions are imposed on the design information space. Many objects captured in n-dim represented artefacts, designers and users, and many of the relationships captured and reasoned about could be regarded as affordances.

The DRed project at Cambridge EDC has for the past seven years been researching the capture of information to support individuals and teams in their design thinking and communication. The research is partially supported by, and conducted in close collaboration with, Rolls-Royce (RR). It has been successful in embedding, arguably for the first time, a rationale tool based on Rittel's seminal IBIS concept [5] in the engineering design culture of a major company [6]. IBIS simply creates a map of the issues or questions being addressed, each linked to alternative proposed answers, which in turn are linked to arguments for or against them. DRed 1.0.8 has been installed as part of the standard PLM tool-set on the RR technical PC network since November 2005. Main areas of application are the root cause analysis and management of aerospace in-service events, design problem analysis, and the generation and selection of design solutions. One of the key factors in the twin success of both producing a usable tool and introducing it into the collaborating company, was a conscious decision from the outset to follow the approach pioneered by n-dim [7]. This is one of evolutionary software prototyping, using a combination of scripting and compiled languages, of a graph-based modelling platform seamlessly supporting both descriptive and prescriptive stages of research, and eventual tool delivery. Equally important was the identification and solution of the technical problems that beset and eventually overcame n-dim. Chief among these was the ambitious complexity and insufficient stability of the n-dim platform itself. Rather than using widely used software components, n-dim defined its own prototype-based object system (BOS) and programming language (stitch). Eventually it was decided to re-implement the n-dim platform from scratch, using a combination of python and java programming languages [8], which was obviously a major task. Despite reaching the beta release stage, this sadly proved unsuccessful. A further hindrance of the n-dim architecture was its reliance on its own dedicated database management system. It has always been a highly difficult task gaining permission for such systems supporting research software to be installed on servers of industrial companies, and this difficulty was further exacerbated by the widespread trend in the 1990s towards out-sourcing of corporate IT provision [6]. DRed succeeded in solving the problems of complexity, stability, and reliance on a database. The need for a database is obviated by its use of an existing open source graph editing tool, storing information in sets of planar graphs, each a plain text file, and interlinked by suitable *bi-directional* hyperlinks. Thus arbitrarily large graph based information spaces can thus be stored in standard corporate file and intranet servers, document management and PLM systems, with very little barrier to tool adoption. Graphlet [9] was primarily chosen as the underlying graph editor because of its use, like n-dim of a flexible scripted graphical user interface overlying an efficient compiled graph processing substrate. This has proved to be an excellent choice being small, robust, and having supported all subsequent core software development so far in pure scripting, with no need to risk stability by modifying the compiled substrate.

In a recent paper [10], Maier and Fadel state: "The central idea of Affordance Based Design is that design is the specification of a system structure that possesses certain desired affordances in order to support certain desired behaviors, but does not possess certain undesired affordances in order to avoid certain undesired behaviors." Further, they state: "In affordance based design, the first task for the designer(s), is to determine the artifact-user affordances that the artifact should have and not have." Thus affordances are specified as an indirect way of specifying what is *really* desired and undesired: the behavioural relationships between artefacts and users. An obvious question is "Why not specify the behavioural relationships directly?". This is not a new idea. The concept of graphical mapping of useful and harmful actions or effects between artefacts, or between artefacts and users, was published as part of a patent application filed in 1997 by the TRIZ vendors Invention Machine Corporation [11]. It was then implemented in their TechOptimizer software, later renamed Goldfire Innovator. These diagrams are a specialised form of concept map [12]. In RR, effective use has been made of them in engineering design for some years, generally in the context of TRIZ workshops convened to address a particular problem [13]. They are generally created in Microsoft PowerPoint with the aid of a simple template and referred to as Functional Analysis Diagrams or simply FADs. A simple example FAD captured by Knott in a RR TRIZ workshop investigating the design of dovetail joints between turbine disks and blades, is shown in Figure 1. The normal practice is to define artefacts, users and other resources using nouns, and the behavioural relationships using verbs. Useful relationships are solid green lines and harmful relationships dashed red.

Maier and Fadel's above definition of ABD now begs a question. If the desired and undesired behavioural relationships are specified using FADs, how can a system structure be designed possessing the correct affordances so as to support the desired behaviours and avoid the undesired ones? Since both IBIS and FADs had separately proved effective in RR a natural thought was to consider using them together. A desired behaviour in a FAD would be bi-directionally linked to the top level issue of a separate DRed IBIS chart, containing a generative design question asking what known artefacts might show the recipient an affordance that would provide the desired behaviour. As a trivial example, consider the design of a suitcase. The ABD process might record that “the suitcase affords holdability to the user”. The equivalent in DRed would be a FAD with two block elements “user” and “suitcase” linked by the useful directed behaviour relation “holds”. This relationship would itself be linked to the root issue of a separate IBIS chart, which might be worded “How can the suitcase afford holdability to the user?”. Well it *could* be worded like that, but it sounds terribly clumsy to the authors. We might word it “How to the allow the user to hold the suitcase?”, which in classical IBIS fashion would be linked to all alternatives thought of, such as “handle”, “shoulder strap”, “towing strap” etc. Each of these would be linked to all arguments thought of for and against.

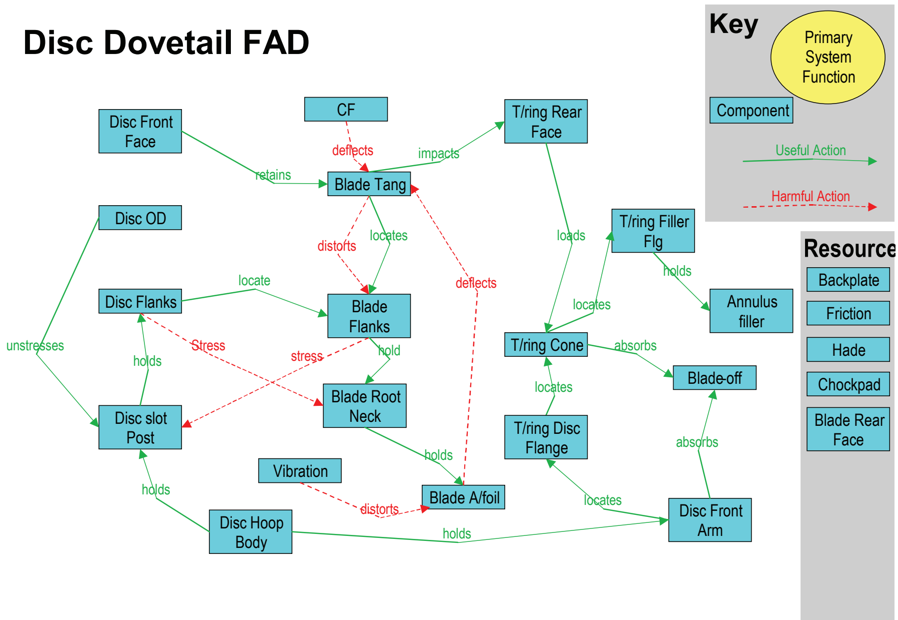

Figure 1. Example FAD created in MS PowerPoint

The above reasoning seems to suggest that the concept of linking IBIS charts to FADs is worth investigating as a potential alternative basis of a unifying, relational design method. But how might it be tested on live design problems? At ICED’07 we reported an extension to DRed, extending its bi-directional hyperlinking to and from bookmarked locations in MS Office documents [14]. This could be used to link PowerPoint FADs to DRed charts in the manner suggested. However, PowerPoint is a far from ideal tool for creating FADs, particularly in live workshop capture situations where streamlined functionality is critical, so it was decided to extend DRed to do the task natively. This was a relatively simple task, simply requiring the definition of two new element types, *Block* and *Relation*. Early trials in RR of the new DRed FAD capability on “live” design projects appeared positive, including application to early design of free stream tidal power stations, and understanding the issues of a novel bearing arrangement in a gas turbine. However, for the integrated design of large, complex systems such as gas turbines, the limitation in size of a FAD model to what can be laid out legibly on a single chart is a severe restriction. DRed’s existing tunnel links, which have proven highly effective at distributing large IBIS graphs over a multitude of charts [6], are much less effective for FADs, as the block–relation–block triples portrayed get split across pairs of linked charts, greatly hindering comprehension. Hence the key research question to be solved for linked FAD/IBIS models to be a viable method for complex system design, and the one addressed in this paper, is: “How can comprehensible and comprehensive integrated FAD models of complex engineered systems be created, managed and linked to IBIS charts to support design?”

2 DRED 2.0: BLOCKS, RELATIONS, DECOMPOSITION & TRANSCLUSION

Previous publications concerning the first generation DRed tool (e.g. [6] [15]) have focussed on the capture and communication of "Design for X" knowledge such as manufacturing considerations and on problem diagnosis, so these uses will not be revisited here. Instead this paper will explore how DRed has been extended in its second generation release to capture and communicate behavioural and functional design knowledge. It is important to emphasize the contrast between the research reported here and mainstream ontological approaches to the representation and capture of design reasoning [16]. DRed's ontology or information model is hard-coded into the software, and the research approach aims progressively to improve user benefit and breadth of application, with the absolute minimum of added complexity. This is because the success of the tool in industrial application has been strongly linked to its ability to capture and communicate graphically the concepts and relationships of design thinking with minimal learning curve and cognitive overhead [6]. In ontological design capture approaches, sophisticated domain specific classes, attributes, constraints and rules are defined externally, using standard languages such as OWL and SWRL (e.g. [17], [18]). The aim is to maximize the formality of the information captured, enabling more sophisticated computational processing of that information. However if users are not effectively shielded from extra cognitive load due to this complexity, it may hinder practical application. While design capture in DRed is deliberately done with a simple, generic ontology, natural language processing techniques may be applied with domain ontologies to perform off-line semantic mark-up of documents [19]. This has the potential to improve context specific retrieval of past rationale and other design information.

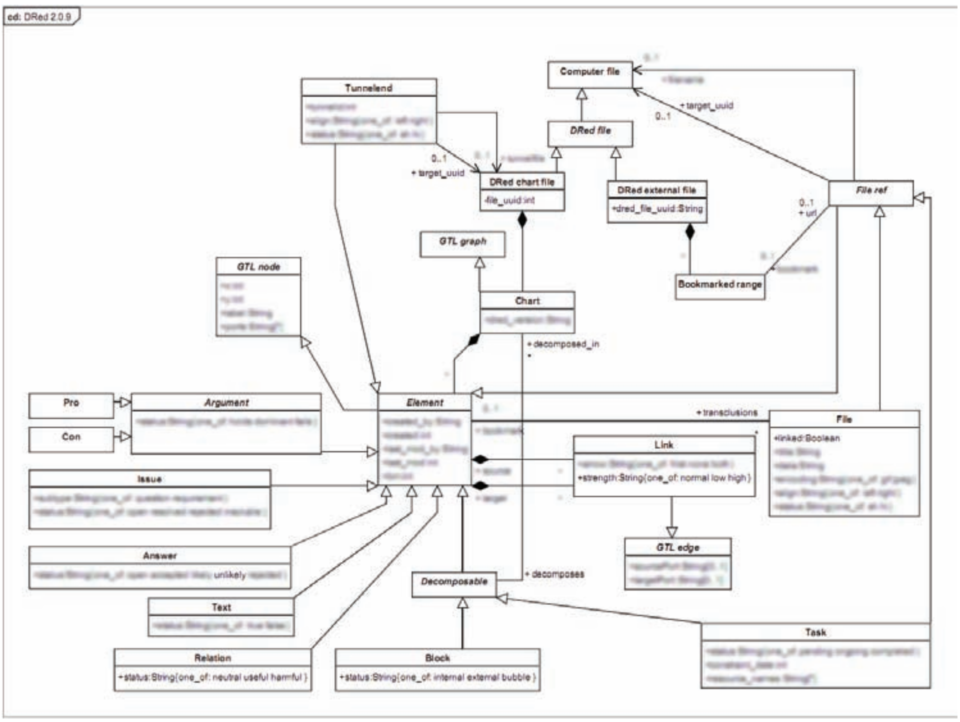

Figure 2. The DRed 2.0.x Information Model

The file format of DRed's plain text .dre files is Graphlet's GML [20], with various additional attributes defined for Graph Template Library (GTL) *graph*, *node* and *edge* objects. Figure 2 displays a UML class diagram [21] showing the complete set of object classes and relationships of DRed v2.0.x and the user-defined GML attributes that implement them. All GML attributes are stored as text strings. Deliberately blurred are the attributes and relationships present in the v1.0.9 information model that has been the standard version in use in RR for the past 3 years. New terms that have been added since v1.0.8 are not blurred. All elements in the v1.0.8 class diagram remain in v2.0.x, apart from the *tunnelnum* attribute of the *tunnelend* class. *Tunnelends* exist in pairs of circular icons that are mutually referencing hyperlinks, and are labelled with generated small integers that are unique per *chart*. These are known as tunnel links [22]. An improvement in v2.0.x is that these id numbers have been made identical for both ends of the tunnel, so the redundant *tunnelnum* attribute has been removed.

The new instantiable element classes in v2.0.x to support FAD modeling are *block* and *relation*. In FADs, *blocks* represent product structure elements, users or general resources, while *relations* represent useful and harmful effects or behaviours. These elements can also be used for general purpose concept mapping. Note that *relation* is a subclass of *GTL node* rather than *GTL edge*, as might be expected. This is because *relations* can be many to many, whereas a graph edge is by definition one to one. Thus a behavioural relationship between *blocks* is represented by two unlabelled *links* with an intermediate *relation* node defining the behaviour. The use of *blocks* and *relations* for FAD modeling will be demonstrated by example in the following section.

In addition to the original tunnel linking, v2.0.x now supports two additional forms of bi-directional link, that have been implemented in other graph based design hypermedia tools such n-dim [4] and Compendium [23]. These are *decomposition* and *transclusion*. What is believed to be new and beneficial is the combination of these two concepts with tunnelling, and together they form the basis for manageable large scale file-based hierarchical FAD modelling. Decomposition is a simply the concept of a whole graph defining the contents of a single node in another graph. For example, Compendium has a specific *Map* node type implementing this concept. Double clicking a map node opens a new window displaying the map or graph contained. As shown in Figure 2, two DRed element classes, *Block* and *Task* are currently defined as *Decomposable*. Thus bi-directional links can be made between a block or task located in one DRed chart and another whole chart decomposing it. In our current implementation a decomposable element is allowed to be decomposed in more than one chart, but a chart is only allowed to decompose a single element. Transclusion was introduced by Nelson as a core concept of Xanadu, the original hypertext project [24] [25]. Paraphrasing the definition in wikipedia.org, transclusion is "the inclusion of part of a document into another document by reference … the reference serving to link both documents". The first limited transclusion facility added to DRed was the ability to create bi-directional hyperlinks between *File* elements and any bookmarkable region of MS Office text, spreadsheet, or presentation documents [14]. A graphical image of the bookmarked MS Office entity is automatically captured via the Windows clipboard and displayed in the file element. A standard MS Office hyperlink is created, pointing from the bookmarked region to the DRed chart and specific file element transcluding it, using a custom dred: URL protocol registered with the operating system. Of course the MS Office hyperlink can only point to a single destination, so a severe limitation is that each region can only be transcluded once. A natural extension was to provide similarly for the transclusion of any DRed element, by hyperlinking to and capturing a bitmap image in a file element. The transcluded element maintains a list of its transclusions (see Figure 2) and the user is able to navigate the list in round-robin fashion. This also effectively removes the limitation of a single transclusion of an MS Office document region. The region can only be hyperlinked to a single DRed *file* element, but that element can then in turn be transcluded as many times as necessary.

An additional enhancement to the flexibility and robustness of all three forms of bi-directional hyperlinking in v2.0.x (tunnelling, decomposition and transclusion) is the application of Open Software Foundation standard Universally Unique Identifiers (UUIDs) [26] for all DRed chart files and linked external files such as MS Office documents. This is reflected in the *file_uuid*, *dred_file_uuid* and *target_uuid* attributes shown in Figure 2. Original DRed tunnelling was based solely on recording relative pathnames between files in a design folder. Thus while the entire folder could be moved without affecting navigation, any internal rearrangement of files would break links. Now arbitrary file renaming and rearrangement is possible, including storage in Product Lifecycle Management systems.

3 CASE STUDY: INTEGRATED PRODUCT STRUCTURE, GEOMETRY AND FAD MODEL OF A 3-SHAFT TURBOFAN ENGINE

In order to demonstrate how the new capabilities of DRed 2.0.x can be applied to support structured design thinking and communication, this section will explain its application to the top down creation of an integrated product structure, geometry and FAD model of a 3-shaft civil airliner turbofan engine. The design of such engines is a classic example of what Pahl and Beitz called *adaptive design* [27], which they defined as follows: "In adaptive design one keeps to known and established solution principles and adapts the embodiment to changed requirements. It may be necessary to undertake original designs of individual assemblies or components. In this type of design the emphasis is on geometrical, analytical (strength, stiffness etc), production and material issues." Each successive generation of engine is consciously adapted from its predecessors, in response to changes in specification and developments in technology. Hence it would be a highly beneficial aid to both individual and team design thinking to have a hierarchical model of the previous generation engine, structured according to the familiar product architecture, allowing the browsing not just of part and assembly geometry, but also useful and harmful behavioural relations and linked rationale justifying geometry and material choice decisions.

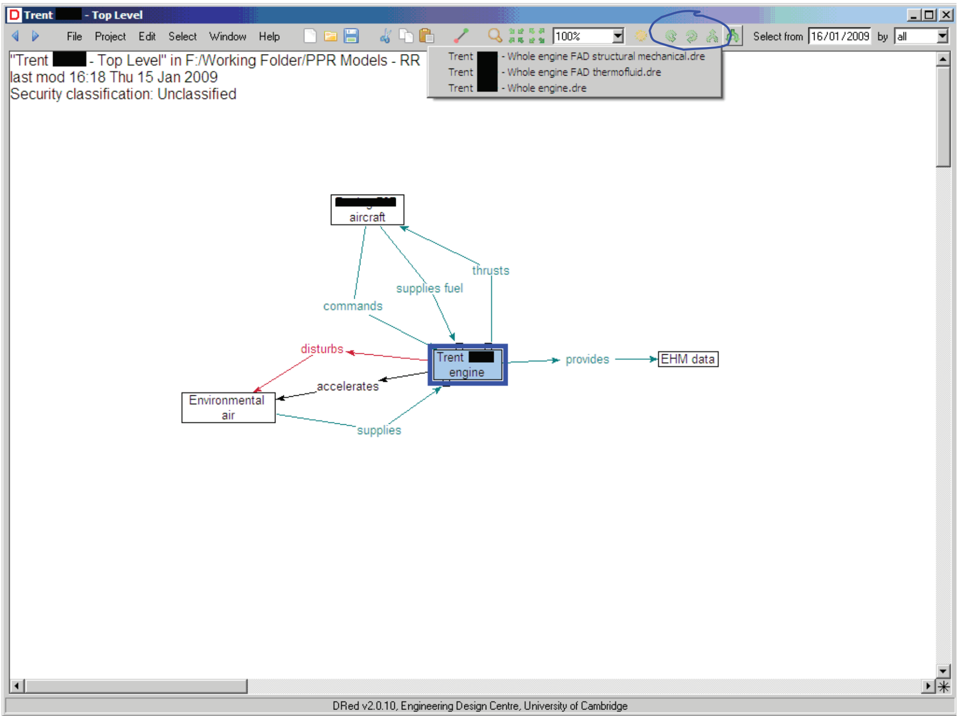

Figure 3. Top Level Context Diagram

The top level of such a model for the Trent engine architecture is shown in Figure 3. The engine is represented by a single block of status internal (internal signified by its grey rather than white fill). The engine has behavioural relationships with three external *blocks* (external signified by white fill), namely the aircraft, the environmental air and the engine health monitoring data. *Useful relations* are that the air supplies the engine, which thrusts the aircraft and provides health monitoring data. The aircraft commands and supplies fuel to the engine. There is also a *harmful relation* that the engine disturbs the environmental air, and a neutral relation that it accelerates it. *Useful relations* are coloured green, *harmful relations* red, and links connected to relation elements are automatically assigned the same colour as the element. The most noticeable change of the user interface of DRed v2.0.x as compared with v1.0.8 is the group of four buttons on the toolbar, circled in Figure 3. The left hand pair allows the navigation of transclusions, and they are mapped also to keyboard shortcuts consisting of the control key plus the left and right arrows. The right hand pair allows the navigation of decompositions, and they are mapped also to keyboard shortcuts of the control key plus the up and down arrows. With the Trent engine block selected (signified by the thick surrounding border to the block), clicking the rightmost of the four buttons (or control key plus down) pops up the clickable menu shown of three chart files defining the next level of decomposition of the engine. Two of these are FADs, one for thermofluid relations and the other for structural and mechanical. The third, “Trent XXX Whole Engine.dre”, is clicked and the result is the display of this chart in a new window as shown in Figure 4.

Figure 4 shows a general assembly drawing of the whole engine, as a bitmap image displayed in a *file* element. As compared with the top level chart in Figure 3, it has an extra line in the title block displayed in the top left corner of the chart. This additional line: *Decomposes “Trent XXX engine” in “Trent XXX Top Level.dre”* displays the other end of the bi-directional decomposition link .that was navigated from the chart in Figure 3. Clicking the second rightmost of the group of four buttons (or control key plus up arrow) would return to select and display the block decomposed. The nineteen *internal blocks* in Figure 4 define the next level of the hierarchical decomposition of the engine.

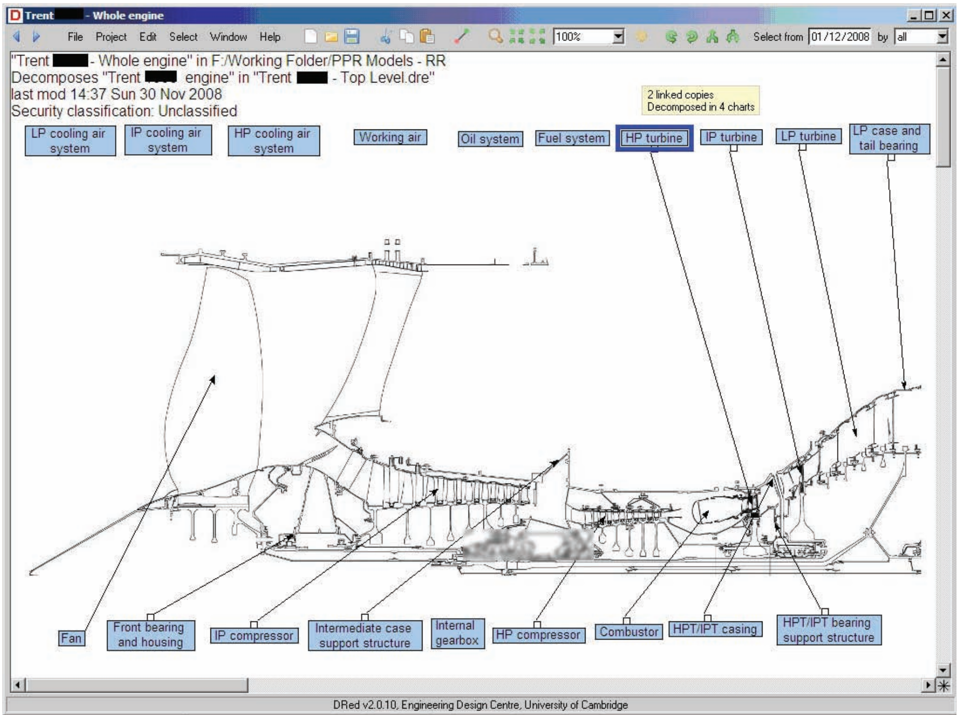

Figure 4. Whole Engine Assembly Geometry and Product Structure

Where the sub-assembly defined by a block is identifiable in the assembly drawing, a link arrow pointing from the block is anchored to the location in the drawing. “Working air” is a block representing an important entity not represented in the bill of materials product breakdown. Selected is “HP turbine”, and above it is the informative tool-tip that pops up when the mouse cursor enters this block. It says that the block has two linked copies and is composed in four charts. *Linked copy* is a synonym for transclusion, used in the DRed user interface as a more descriptive term for users unfamiliar with the hypertext research literature terminologies. With this block selected, clicking either of the transclusion navigation buttons (or control key plus left or right arrow) will enable a round-robin traversal of this block and its two transclusions in other charts, going either backwards or forwards respectively through the list.

Going forward one element in the transclusion list opens the window and chart shown in Figure 5. This is the structural and mechanical FAD of the whole engine, which as the title block shows, also decomposes block “Trent XXX engine” located in “Trent XXX Top Level.dre” The transcluded image of “HP turbine” is now highlighted. Transcluded images of DRed elements are clearly distinguished from their source by the imposition of a small red triangle in the top left corner of the image.

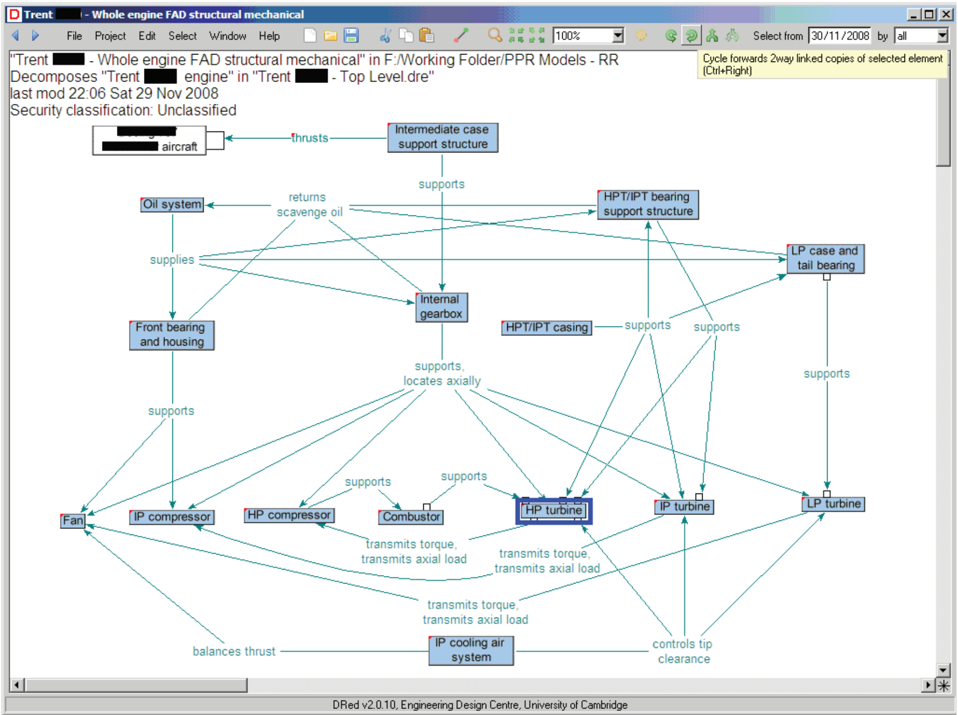

Figure 5. Whole Engine Structural-Mechanical FAD

Clicking the same button again navigates to the second transclusion of HP turbine, this time located in the thermofluid FAD of the whole engine, which is shown in Figure 6. Note that in Figure 6, one of the tiny white squares surrounding the transclusion of block “Working air” has been selected. These squares have also appeared in earlier figures without previous mention. It can be seen that some arrow links end directly on blocks, while others end on these small white squares on the edge of the blocks. This signifies that some behaviours originate from, or apply to a whole block, whereas others should burrow inside the block to terminate at a specific location in the block’s decomposition. This is similar to the behaviour of flows in a set of levelled hierarchical dataflow diagrams, such as in a Yourdon Structured Analysis of an information system [28]. In DRed, this “burrowing” is provided by the tunnel links described in Section 2. The white squares are simply the way normal tunnel links are displayed when they link elements in two charts that share a decomposition relationship. The normal circular shape of the pair of linked tunnelend icons is changed to square, and the one at the higher level in the decomposition is reduced in size by a factor of about three. This enables it to be aligned unobtrusively on the edge of the block decomposed in the chart of the tunnel destination. Thus the tunnelend appears as a port into the decomposed block, linking to a particular sub-block in the decomposition.

Take for example the highlighted tunnel adjacent to “Working air” in Figure 6. It links to the useful relation “extracts power” which is performed by “HP turbine”. The “Working air” block is decomposed in the chart in Figure 7, which shows that “Working air” consists of various parts, such as “Fancase air”, “Bypass air”, “IPC annulus air”, “HPC annulus air” etc. While browsing the chart in Figure 6, a user might be interested in knowing what part of the “Working air” the “HP turbine” extracts power from. The answer is obtained by double clicking the highlighted tunnel, causing the tunnel to be traversed into the decomposition of “Working Air” in Figure 7.

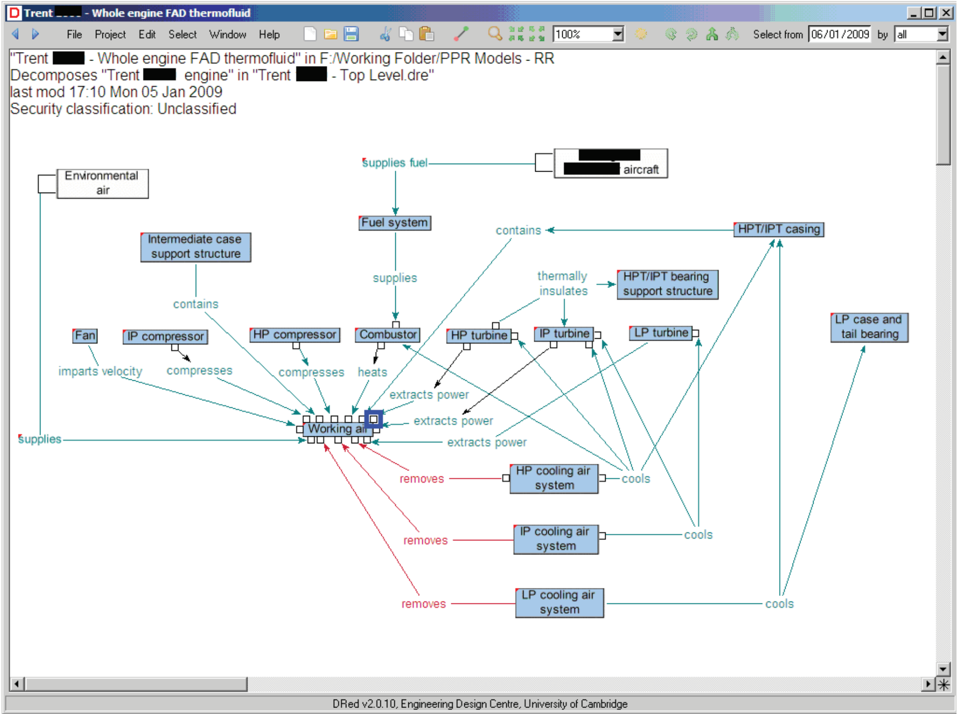

Figure 6. Whole Engine Thermofluid FAD

The far end of the tunnel is now highlighted as shown, this time represented by a normal size but square tunnel. This links to a transclusion of the “extract power” relation from the level above, transcluded so that the relation is visible on both decomposition levels. This then links to the part of the “Working air” from which power in extracted, which it can be seen is “HPT annulus air”. There is still a need to show in the chart in Figure 7 what product structure element on the level above extracts power from the “HPT annulus air”. This is done by aligning with the tunnelend a *block* of status *external* (signified by white background fill), showing that the destination is “HP turbine”. DRed is able to create these appropriate external blocks to document decomposition tunnels automatically.

In a Yourdon Structured Analysis hierarchical dataflow model of properly designed information system [28], there is never a need for flows to cross link the hierarchy, so it is not allowed. That is not the case for a hierarchical FAD model of a complex product such as a gas turbine. As shown in the example so far, Yourdon-style hierarchical flow-down allows the mapping of a FAD model onto the standard product structure that is well known by the company’s engineers, expressing clearly the primary functional relationships of assemblies and sub-assemblies. However, important behavioural relationships may exist directly between any pair of blocks, at any levels in the product breakdown.

The hierarchical FAD model must may able to cater for this. It is the same need that was addressed by the *flat space of objects* that was one of the core ideas of n-dim [4]. Conventional tunnel links carrying a relationship between any pair of blocks provide exactly this capability. Take for example the combustor and working air blocks in the whole engine thermofluid FAD of Figure 6. The only thermofluid functional relationship between these at the whole engine level is that the combustor heats the working air. Going down into the thermofluid decomposition of working air in Figure 7, it can be seen additionally that some part of the combustor also contains the combustion chamber air, and that some part of the combustor also slows and increases static pressure of the HP compressor annulus air. These additional relationships are carried by tunnels cross-linking the decomposition hierarchy, and thus the tunnelends adjacent to the combustor external block appear circular rather than square. If an engineer wanted to find what part of the combustor was performing each function, double clicking either tunnel end would open the thermofluid FAD decomposition of combustor, showing that the first function is performed by the casing and the second by the diffuser.

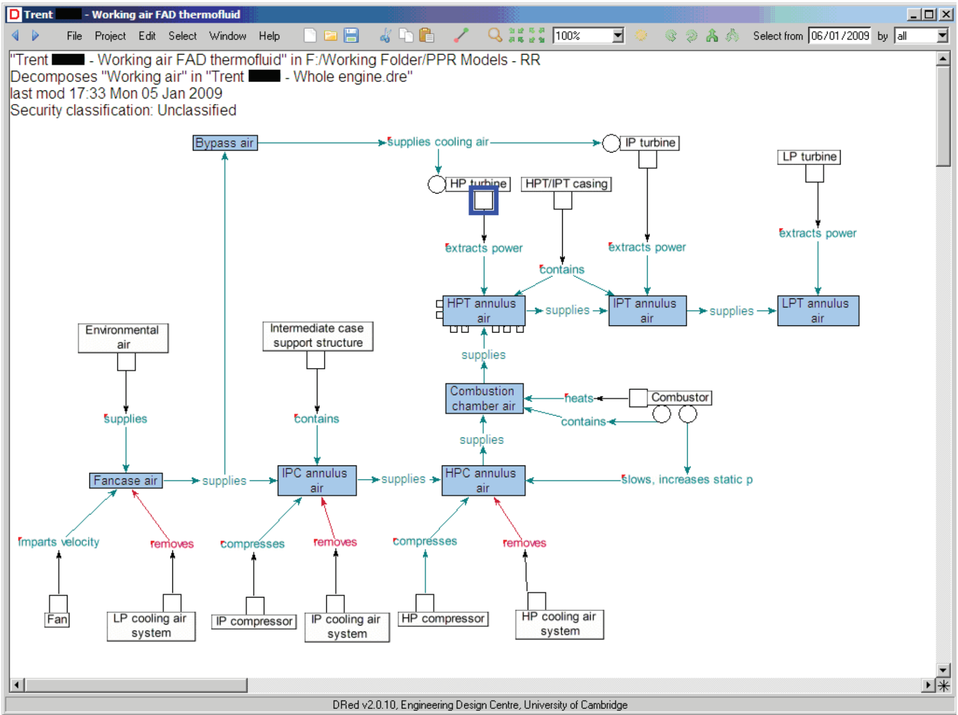

Figure 7. Working Air Thermofluid FAD

4 CONCLUSIONS AND FURTHER WORK

This paper has demonstrated by example how a relatively simple graph based software tool can be applied by individual designers and teams to capture understanding of desired and undesired behavioural relationships in the design of complex systems. The software has been researched and developed in close collaboration with engineers is industry and already tested with promising results in practical use on complex systems design. The approach taken was to extend an existing graph based software tool for IBIS based design rationale capture that has been in widespread use in the collaborating company for several years. In the example presented in this paper, the upper levels of an integrated graphical functional, assembly geometry and product breakdown structure of a gas turbine is shown. Work is continuing on the lower levels of this decomposition, down to individual design features and parameters, and performed in such that way that QFD2 House of Quality matrices [29] linking functions to product elements may be exported from the linked set of FADs as a useful by-product. Work also continues to optimise the user interface functionality to make hierarchical FAD modelling as easy and natural as possible for users, which is particularly important for successful use in facilitating group design discussions.

ACKNOWLEDGEMENTS

The research reported in this paper was partly undertaken through the "Immortal Information and Through-Life Knowledge Management: Strategies and Tools for the Emerging Product-Service Paradigm" (KIM) project, funded by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/C534220/1. It was also partly supported by the UK Technology Strategy Board through the HIPARSYS project and Rolls-Royce plc through the UTP for Design.

REFERENCES

[1] Maier J.R.A. and Fadel G.M. Affordance based design: a relational theory for design. *Research in Engineering Design*, in press, DOI 10.1007/s00163-008-0060-3.

[2] Gaffney E.S., Maier J.R.A. and Fadel G.M. Roles of Function and Affordance in the Evolution of Artifacts. In *International Conference on Engineering Design, ICED'07, Paris, France* August 2007.

[3] Gibson J.J. The theory of affordances. *The ecological approach to visual perception*. 1979 (Houghton Mifflin, Hopewell), pp.127–143.

[4] Subrahmanian E., Konda, S.L., Levy, S.N., Reich, Y., Westerberg, A.W., and Monarch, I.A. Equations aren't Enough: Informal Modeling in Design. *AI EDAM*, 1993, 7(4), pp.257-274.

[5] Kunz W. and Rittel H.W.J. Issues as elements of information systems. Working paper 131. Center for Planning and Development Research, Berkeley; 1970.

[6] Bracewell R.H, Wallace K.M. Moss M., Knott D.S. Capturing design rationale *Computer-Aided Design,* in press, DOI: 10.1016/j.cad.2008.10.005.

[7] Subrahmanian E., Reich Y., Konda S.L., Dutoit A., Cunningham D., Patrick R., Thomas M., Westerberg A.W. The n-dim Approach to Creating Design Support Systems. In *Proceedings of DETC/DTM 1997*, DTM-3873. 1997.

[8] Cunningham D., Subrahmanian E. and Westerberg A. User-Centered Evolutionary Software Development Using Python and Java. In *6th International Python Conference*, San Jose, October 1997, pp.1–9.

[9] Himsolt M. Graphlet: Design and implementation of a graph editor. *Softw. Practice Exp.* 2000;30(11), pp.1303-24.

[10] Maier, J.R.A. and Fadel, G.M. Affordance Based Design: Status and Promise. In *IDRS'06*, Seoul, South Korea, 2006.

[11] Devoino I.G., Koshevoy O.E., Litvin S.S. and Tsourikov V. Computer Based System For Imaging And Analyzing An Engineering Object System And Indicating Values Of Specific Design Changes. United States Patent 6056428, filed 1997.

[12] Novak J.D., Concept maps and Vee diagrams: two metacognitive tools to facilitate meaningful learning. *Instructional Science,* 1990, 19(1) / January, pp.29-52.

[13] Knott D.S. The Place of TRIZ in a Holistic Design Methodology. *Creativity and Innovation Management,* 2001, 10(2).

[14] Bracewell R.H., Gourtovaia M., Wallace K.M. and Clarkson P.J. Extending Design Rationale to Capture an Integrated Design Information Space. In *International Conference on Engineering Design, ICED'07, Paris, France* August 2007.

[15] Bracewell, R.H., Ahmed, S. and Wallace, K.M. DRed and Design Folders, a Way of Capturing, Storing and Passing on Knowledge Generated during Design Projects, *ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Salt Lake City, Utah, USA, 2004, DETC2004-57165.

[16] Szykman S., Sriram R.D., Regli W.C. The Role of Knowledge in Next-Generation Product Development Systems. *ASME Journal of Computing and Information Science in Engineering,* 2001, 1(1), pp.3-11.

[17] Borgo S., Carrara M., Garbacz P. and Vermaas P.E. A Formal Ontological Perspective on the Behaviors and Functions of Technical Artifacts. *AI EDAM*, 2009, 23(01), pp.3-21.

[18] Kim K.-Y., Chin S., Kwon O. and Ellis R.D. Ontology-Based Modeling and Integration of Morphological Characteristics of Assembly Joints for Network-Based Collaborative Assembly Design. *AI EDAM*, 2009, 23(01), pp.71-88.

[19] Kim, S., Bracewell, R.H. and Wallace, K.M. Answering Engineers' Questions Using Semantic Annotations. *AI EDAM*, 2007, 21(2), pp.155-171.

[20] Himsolt M., GML: Graph Modelling Language. http://www.infosun.fmi.uni-passau.de/Graphlet/download/misc/GML.ps.gz, 1996.

[21] http://en.wikipedia.org/wiki/Class_diagram

[22] Bracewell R.H. and Wallace K.M. A Tool For Capturing Design Rationale. *ICED'03, Stockholm, Sweden,* August 2003.

[23] Buckingham Shum S.J., Selvin A.M., Sierhuis M., Conklin J., Haley C.B. and Nuseibeh B. Hypermedia support for argumentation-based rationale: 15 years on from gIBIS and QOC. *Rationale management in software engineering*, 2006 (Springer-Verlag), pp.111-132.

[24] Nelson T.H. *Literary machines*, 1981 (Swarthmore, Pennsylvania).
[25] Nelson T.H. Xanalogical Structure, Needed Now More than Ever: Parallel Documents, Deep Links to Content, Deep Versioning and Deep Re-Use. *ACM Computing Surveys (CSUR) Archive,* 1999, 31(4).
[26] http://www.opengroup.org/onlinepubs/9629399/apdxa.htm
[27] Pahl G., Beitz W., Wallace K.M., Feldhusen J., Blessing L.T.M., Grote K.H. *Engineering Design: A Systematic Approach*, 2007, (Springer).
[28] Yourdon E., *Modern Structured Analysis*, 1989 (Prentice-Hall).
[29] Cristiano J.J., Liker J.K. and White C.C. Customer-driven product development through Quality Function Deployment in the U.S and Japan. *Journal of Product Innovation Management*, 2000, 17(4), pp.286-308.

Contact: Rob Bracewell
Cambridge University
Department of Engineering
Trumpington Street
Cambridge, CB2 1PZ
UK
Tel: +44 (0)1223 748567
Email: rhb24@cam.ac.uk
URL: http://www-edc.eng.cam.ac.uk

Rob Bracewell is a Senior Research Associate in the Cambridge Engineering Design Centre. He received his PhD in the design of ocean wave-energy converters and subsequently researched in the fields of mechatronics and advanced robotics. Currently, his main interests are in the creation of software tools, to aid design teams in the generation, selection and realisation of innovative, high performance design solutions.