

EXTENDING DESIGN RATIONALE TO CAPTURE AN INTEGRATED DESIGN INFORMATION SPACE

Rob Bracewell, Marina Gourtovaia, Ken Wallace and John Clarkson

Engineering Design Centre, Department of Engineering, University of Cambridge, UK

ABSTRACT

DRed is a graphical software tool for design rationale capture that, despite essentially still being a research prototype, has proved robust and useful enough gradually to achieve widespread use in an international aerospace company. The main areas of application up until now have been: (1) in the early stages of design; (2) in the root cause analysis and design of solutions to problems occurring while a product is in service.

In order to lay out and navigate freely large interlinked rationales across multiple charts, DRed uses a simple bidirectional hyperlinking approach known as tunnel linking. In this paper we suggest that if it could be made easy for users to create bidirectional hyperlinks between DRed elements and selected locations in a range of *external* document types, then this might support the capture not just of the rationale, but of a unified, easily navigable information space covering the specification, rationale, calculations, design tasks and the emerging product definition.

The objective of this research was to devise a practical way of implementing such a facility, then to explore how integrated design information spaces captured in this way might be structured, and to assess the feasibility of their capture as the design proceeds. To be successful, this will require extending the routine use of DRed from the conceptual into the embodiment design phase. Thus the case study in this paper focuses on the embodiment design of a simple mechanical transmission.

Keywords: design rationale, integrated design models, bidirectional hyperlinks, annotation tools

1 INTRODUCTION

Research into how to capture design rationale has recently progressed from scientific discussions [1], software prototyping and testing [2] [3], to the widespread deployment and successful practical use of such tools, for example Compendium [4] and DRed [5]. Both of these tools are developments of the well-known IBIS concept. In use in a multi-national engineering company, DRed has been found (1) to help designers view, clarify and structure their design thinking; (2) assist with managing design tasks; and (3) capture design rationales as they are created, thereby reducing the need for subsequent written reports. Figure 1 displays a typical DRed chart displaying the issues, answers and arguments of the rationale for a piece design addressing an in service problem with a gas turbine engine. DRed employs a bidirectional hyperlinking approach known as *tunnel linking* to allow single large rationale graphs to be distributed over many documents and subsequently navigated freely in any direction from any starting point. The rationale document files reside in the same repositories and folders as the other files created during the design process. These files are of a variety of different types, containing product specifications, geometry definitions, analysis and optimisation results, emails, presentations, meeting minutes and so on. Hyperlinks may be created from a DRed element to related files, with an optional bookmark string to indicate the location of interest in the target document. In the sole but common case of Microsoft Word documents stored in mounted file-systems, a selected region defined as a bookmark in the document, and referenced by a DRed File element, can be navigated to and highlighted automatically when the File element was double clicked. Thus DRed can be classed as a type of external annotation tool for digital documents.

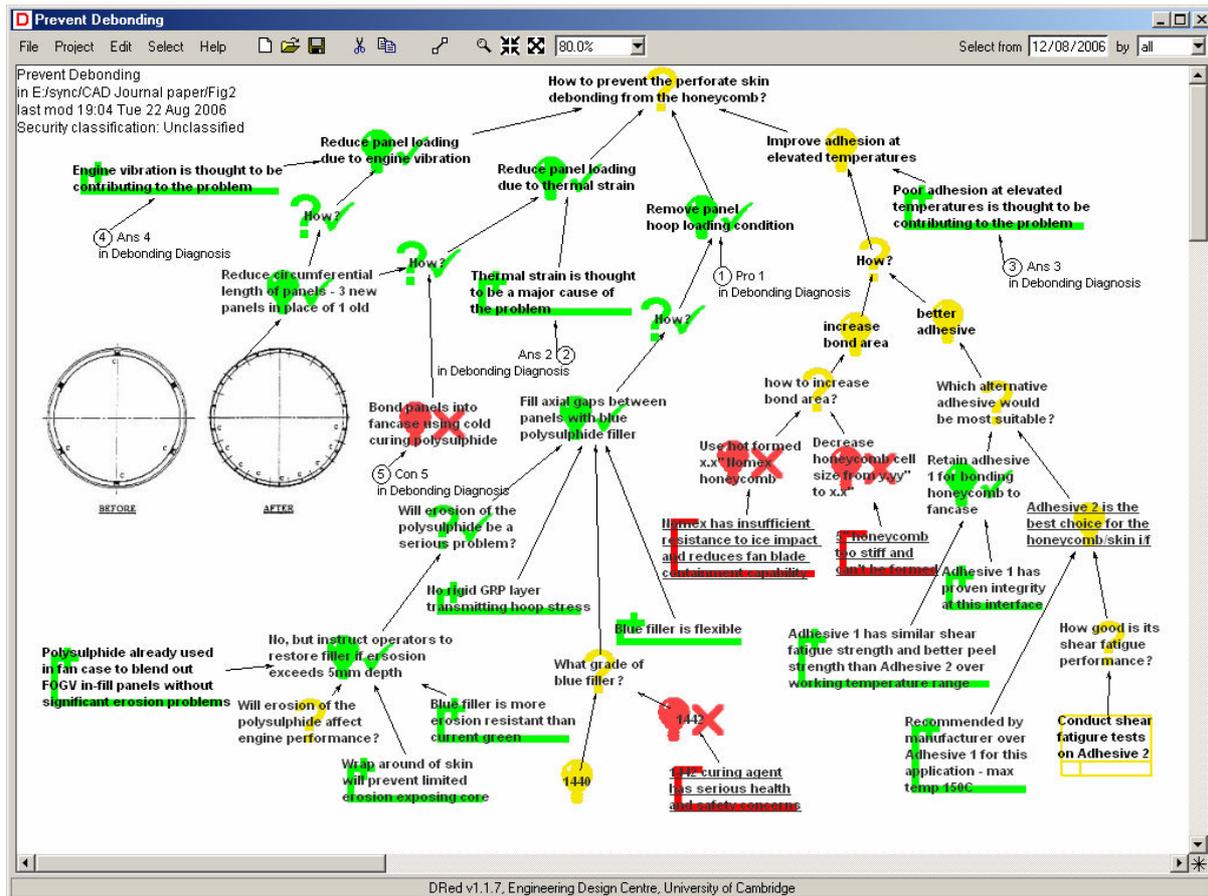


Figure 1. Typical Dred chart capturing design rationale

There has been considerable research in recent years into software tools for annotation, both for general use and specifically to support engineering design collaboration. For example, Longueville reports on the evaluation in industry of a prototype collaborative annotation tool used by a group of EADS engineers [6]. Cabanac lists and compares the features of twenty commercial, free and research prototype annotation systems [7]. Where the annotations are stored externally to the annotated document, most recent tools rely on Web or Semantic Web technologies for location referencing and mark-up. These technologies range from simple HTML facilities through more recent and capable standards such as SVG, XML, Xpointer and RDF [8]. In an ideal world all design documentation would be stored in formats amenable to this approach, and one day this may well be the case, however, in the real world of today things are not so simple. Consider, for example, a seemingly straightforward need to create and navigate relative-path hyperlinks between anchored locations in documents in a given folder. This capability has been supported by HTML since the very inception of the World Wide Web, but it still presents severe problems for office and engineering document types commonly used in design processes. The capability was introduced to Windows in Microsoft Office 2000 in the form of the ActiveX Hyperlinking API [9] but due to its complexity this has been rarely used outside Microsoft and in recent hyperlinking software such as OneNote 2007 by Microsoft it has been abandoned.

Sidewalk [10] is a rare contemporary example of an annotation research project addressing the practical problems posed by today's common office applications. The project has created the SPARCE middleware framework for superimposed applications [11],[12]. SPARCE is an extensible architecture, providing plug-ins, which add a toolbar and/or menu entries to Microsoft Word, Excel, PowerPoint, Media Player and Internet Explorer, and also to Adobe Acrobat. When invoked, these copy a Uniform Resource Identifier (URI) string [13], that references the document and the current highlighted selection within it, to the Windows clipboard. The URI is of a custom scheme prefixed *sparce:*. Most modern hyperlink-aware applications deal with such custom URIs in the following way: when a hyperlink is clicked, they pass the string to the handler registered with an operating system (Windows in this instance) for that URI type. The *sparce:* handler uses COM scripting to load the

target document, launching the application first if necessary, navigate to and highlight the selection referenced by the URI.

An important feature of design information spaces is that they continually change as the design progresses through different stages of the design process. The number of electronic resources relevant to a project continually increases. Each document, whether text, drawing, calculation result or a model, is created with a certain purpose in mind in response to a certain need, such as to clarify and/or support design decisions. A designer drives the process of creating new sources of information and retrieving existing ones; this gives a unique opportunity to engage a designer into a process of creating connecting paths or links throughout the design information space. The facility to handle *sparse*: URIs has recently been added to DRed, and it provides a most convenient way of referencing precise selections in a variety of document types that are commonly used in design. It is particularly useful in linking to read-only reference documents, as write access to the document is not required to define the target of the link. However, if this linking approach is used with documents created as part of the design process, the future reader of the document is at disadvantage, since it is not clear whether a document is linked into the DRed information space at all, and if it is what the sources and targets of the links are. Unlike the tunnel links used within DRed, the SPARCE-type external hyperlinks are *unidirectional*, leading out, but not back in. We suggest that this hinders the free navigation of the design space, which is necessary for the rapid appreciation of the reasoning behind the choices made in the design.

This paper explores the proposition that if it could be made easy for users to create *bidirectional* hyperlinks between DRed elements and selected locations in external documents, then this might support the capture not just of IBIS-style rationale, but of a single integrated information space covering the whole design process. The following section describes how such a facility has been implemented. A convincing demonstration of the above proposition will require extending the routine use of DRed from the conceptual into the embodiment design phase. Thus the succeeding case study presents the use of the newly implemented DRed facilities, together with Microsoft Excel, to perform and capture a mechanical transmission embodiment design exercise.

2 BIDIRECTIONAL HYPERLINKS FROM DRED INTO OTHER DOCUMENTS

2.1 General approach

The proposed system of bidirectional links between DRed and other project documents is not dissimilar to the HTTP protocol hyperlinks that combine distributed and autonomous web resources into a vibrant world-wide network. The difference is that there will not be any default application, like a web browser, that is responsible for rendering the documents. The editor or viewer that is registered by the operating system for each document type should view the document. Like the SPARCE framework, the method implemented by us is based on the use of COM or some other remote scripting framework for the target applications. However, unlike SPARCE, no purposeful application plug-ins are required, because the whole bidirectional link creation and manipulation process is driven from DRed.

Most users are familiar with the concept of a bookmark in software, particularly as implemented in Microsoft (MS) Office applications. If a document contains a bookmark, it can be opened at the position of this bookmark; bookmarks enable creating shortcuts to specific places in a document, which is a very convenient feature for large documents or those containing heterogeneous information. The level of granularity that bookmarks support depends upon the application. For example, any position on a web page, apart from the space taken by images and other objects, can be bookmarked. In MS Word and OpenOffice.org Writer a bookmark is tied to a range of text, in MS Excel it is possible to bookmark a cell or a range of cells. It is logical to use the familiar concept of bookmarks to enable precise navigation of the design information space. However, some applications (editors and/or viewers) might need to be manipulated through their APIs in order to make them support bookmarks at the right level of granularity.

2.2 DRed as a Target for Hyperlinks

In order to have sufficiently fine granularity to capture the reasoning involved in the design process, bookmarks in DRed are implemented on a level of individual nodes, similar to tunnelling between DRed charts themselves. Each DRed element has a persistent integer ID string that is unique in that

chart, so it is natural to use that as the bookmark. If DRed were to use the MS Windows ActiveX Hyperlinking API, that is mentioned in the previous section, a system call to open a DRed chart *my.dre* with element of ID *my_bookmark_num* highlighted, would in essence look as follows:

path_to_dred\dred.exe path_to_my_chart/my.dre#my_bookmark_num

A standard MS Office hyperlink to a DRed file can have a *#my_bookmark_num* specified, but for DRed to receive the value of the bookmark from the operating system (Windows) it would need to provide a COM server with an implementation of the *IHlinkTarget* interface. Without this, the value of the bookmark is not passed to the target application at all. However, implementing this seemingly simple functionality in DRed would have been far from trivial, as the ActiveX Hyperlinking API is highly complex, supporting not mere hyperlinking but a partial integration of external applications into the MS Internet Explorer GUI. Up to date working code examples proved highly elusive, as did any cases of successful use of the API by developers outside Microsoft. It was then that the discovery was made that the recent MS Onenote public beta test release, which prominently featured extensive new hyperlinking capabilities, was not using ActiveX Hyperlinking either, but instead took the previously described approach used by SPARCE, of registering its own *onenote:* URI scheme and handler. Adopting this approach with DRed was clearly the best option, which is also portable across main modern operating systems. Thus a custom *dred:* URI scheme was created, with the main DRed executable registered with the operating system to handle such URIs. A call to open a DRed file with a particular element selected through this protocol looks as follows:

dred:name_of_calling_application(optional):path_to_my_chart/my.dre#my_bookmark_num

This syntax can be used from any application that supports hyperlinking to custom URI schemes. Every DRed element has a context menu item to copy its URI to the Windows clipboard, ready to be pasted into a hyperlink field of such an external application. If *path_to_my_chart* is an absolute path, it can currently either refer to a locally mounted or UNC addressed server file-system or, for read-only access, it can also address files on *http:* or *https:* web-servers. An added advantage of using the *dred:* protocol is that links to DRed are easily distinguishable from links to any other applications. The left hand half of Figure 2 illustrates such a hyperlink from an MS Excel document to a DRed *File* element.

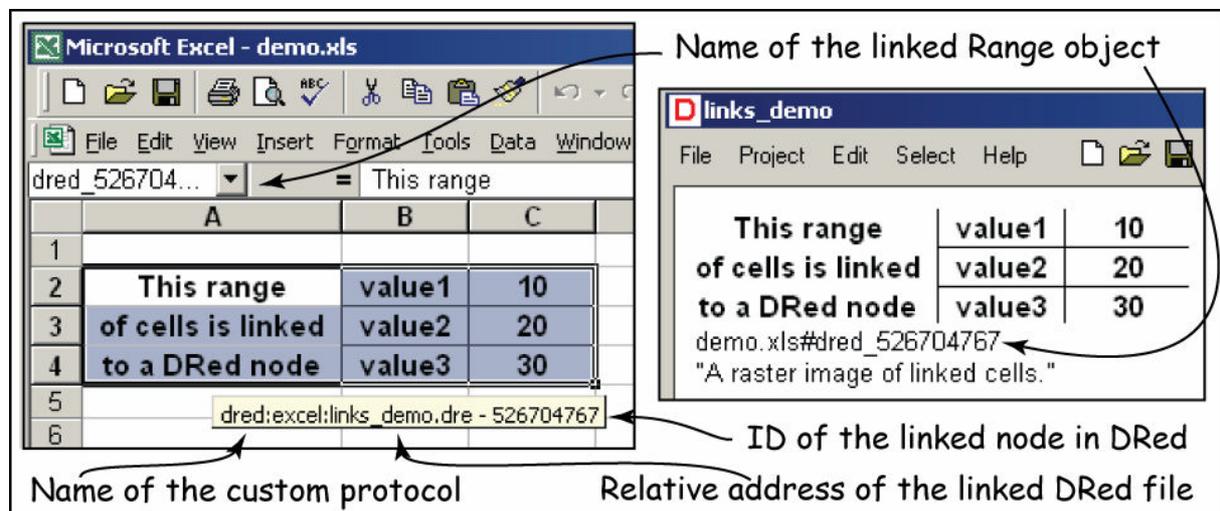


Figure 2. Bidirectional hyperlink between an Excel cell range and DRed

One of the defining principles of DRed is that all tunnel links and hyperlinks between files in a working design folder should be relative rather than absolute, so that they will still work if the folder is moved elsewhere or accessed using a different protocol [5]. The optional *name_of_calling_application* field allows this principle to be maintained for hyperlinks into DRed from files in the design folder. When a *dred:* URI is invoked from an external application, the URI containing a relative path is passed to DRed by the operating system. In order to open the target file, DRed needs to know the absolute location of the external application file from which the hyperlink was invoked. By specifying the name of the calling application, DRed is able to communicate with the running instance of that application, and query the storage location of the document that has the current input focus. This

facility has so far been implemented for a single external application - MS Excel – successfully allowing relative hyperlinks to be followed into DRed from Excel files residing on both mounted file-systems and web-servers.

2.3 Bookmarks for Cell Ranges and Embedded Objects in Excel

In aiming to provide support for the capture of integrated information spaces as design proceeds, it was natural to choose MS Excel as the first target application for bidirectional hyperlinking with DRed. MS Excel is a widely used tool in engineering design for several reasons. First is the worldwide ubiquity of MS Office, so it is conveniently available on the vast majority of personal computers used by engineers to perform ad hoc calculations, simple scripting tasks and to manipulate tables of numerical data. Secondly, the output of specialised analysis software is often available in this format. Thirdly, most modern PC-based parametric solid modelling CAD tools support linking of MS Excel worksheets to part and assembly files, so that parameter names and values defined in Excel and hyperlinked into DRed may also directly drive the CAD geometry.

The right hand side of Figure 2 shows a DRed chart containing a single *File* element referencing bookmark `dred_526704767` in file `demo.xls`, which has been defined as the selected range of cells shown. Remote COM scripting of MS Excel from DRed allows the bookmarked range to be selected and displayed when the DRed element is double-clicked. The default display of DRed *File* reference element is a small icon showing the type of the target file, but they have the alternative of displaying an arbitrary raster image. For MS Excel this option was extended to capture automatically a raster image of the target cell or range of cells. This was also achieved purely through automation COM calls raised from DRed, without adding any Visual Basic code to MS Excel. Figure 2 shows the automatically captured image in the *File* element of the bookmarked cell range, the two mutually referencing each other as a bidirectional hyperlink. In order to avoid the waste of precious chart space, an intelligent trim operation is performed on automatically captured images before they are inserted into DRed. Using the capabilities of the *Imagemagick* library [14], this removes any bounding box and surrounding expanse of single solid colour, closely cropping around the “interesting” part of the selection.

Apart from data in cells, an MS Excel spreadsheet can contain images, charts, equations, drawings and other embedded objects. However, not all of these types of objects can be assigned bookmarks programmatically and/or are able to bear a hyperlink. However, all object types can be grouped with other objects and any group can be bookmarked and bear a hyperlink. Thus, to hyperlink from an MS Excel embedded object into DRed, it is grouped with a small text box with a red letter D; this icon-looking textbox with a transparent background is displayed in the left top corner of the host object’s bounding box (Figure 3). This simple approach allows any object type to be bookmarked, provides a holder for the reverse `dred:` hyperlink URI, and makes hyperlinked objects highly visible.

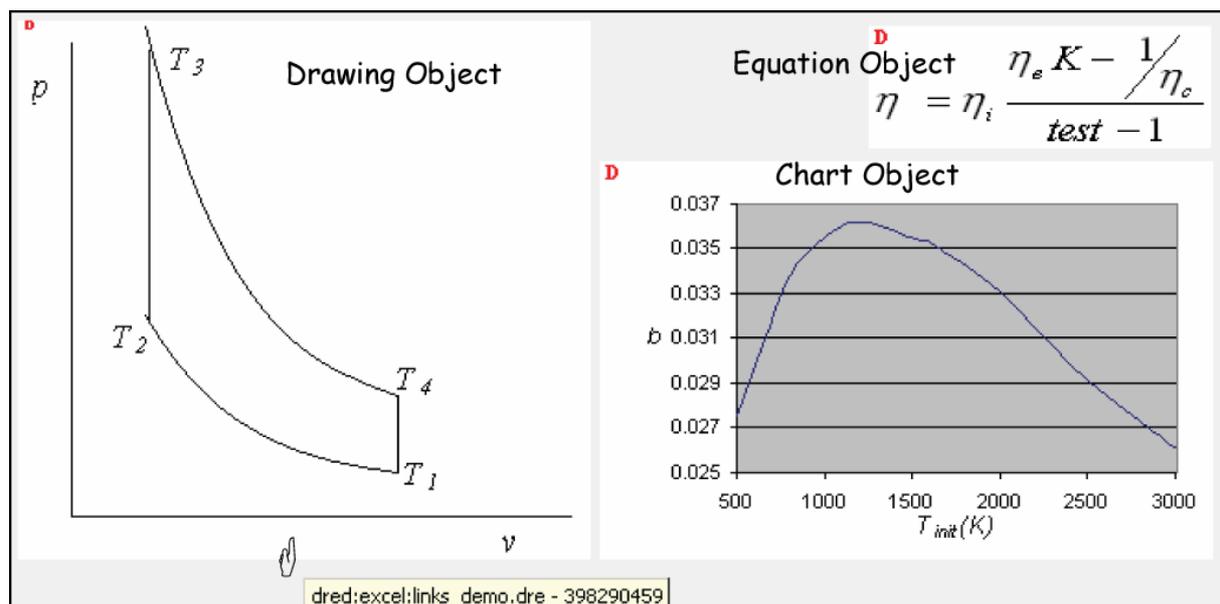


Figure 3. Embedded objects in Excel hyperlinked to DRed

In order to facilitate the rapid, easy creation of bidirectional hyperlinks between DRed and Excel, a new option was added to the main DRed element creation menu (see Figure 3, left menu). The user simply selects the desired hyperlink target in Excel, then right clicks the background of the DRed chart and chooses “Excel Paste”. This: (1) creates a DRed *File* element at the clicked location; (2) links it to the current MS Excel file and selection; (3) links this selection in turn back to the new DRed element; (4) captures a raster image of the MS Excel selection into DRed. To give the user full control over the linking process, all of these steps can be performed manually a step at a time. It is also possible to toggle the creation and deletion of reverse links from an MS Excel object to a DRed *File* element. The captured image can be subsequently updated in DRed if the source changes in MS Excel. If a DRed node, to which a link from MS Excel exists, is cut and pasted between DRed charts, the URI to it in a stored MS Excel file is updated automatically in order to maintain the link.

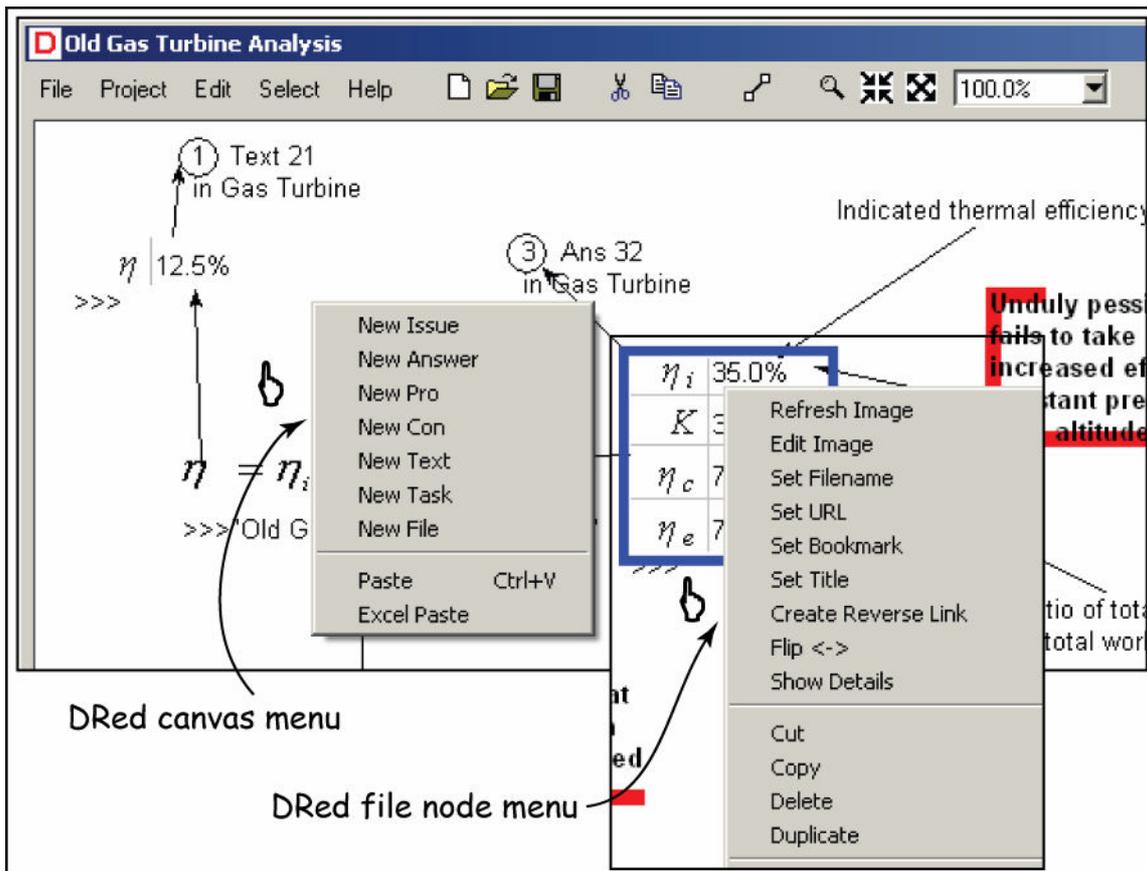


Figure 4. DRed's options for creating and manipulating bidirectional hyperlinks to Excel

3 CAPTURING EMBODIMENT DESIGN AS AN INTEGRATED INFORMATION SPACE

A case study is now presented exploring how the new DRed capability, described in the previous section, might support the conduct and capture of embodiment design activities. The approach taken was to study a video of fellow KIM project¹ researchers from the University of Bath, performing an undergraduate engineering design exercise. They were capturing their design using the conventional approach of a paper designers' notebook, and were performing calculations using a desktop calculator. The aim was to show how the notebook and calculator might be replaced by the combination of DRed and MS Excel, performing the required calculations and capturing and presenting the dependencies the design in an integrated space of information. The exercise involves the design of a lay-shaft sub-assembly that provides a speed reduction power transmission between an induction motor and a hydraulic pump. Figure 5 shows the first page of the student briefing document, which has been imported into MS OneNote 2007. The highlighted “0-->” is a manually created bidirectional

¹ For details, see the Acknowledgements section

hyperlink, using a combination of dred: and onenote: URIs, between the requirement for the pump to deliver 3 l/s, as specified in the briefing document and the location of that requirement in DRed. The participants chose to implement the transmission using toothed belts and grooved pulleys. Figure 6 displays how the design activity of the first 45 minutes of the video, which was originally captured in 7 A4 pages of notebook, could have been performed and captured using MS Excel and DRed. The activity shown focussed on the selection tasks and supporting calculations in choosing a suitable motor, belt types, lengths and widths, and numbers of grooves in the pulleys. Each numerical element on the chart is a selected range of three horizontally adjacent spreadsheet cells, which has been linked into DRed using the “Excel Paste” procedure. The left hand cell of each triplet stores a numerical value or formula, the middle one the units, and the right hand one a descriptive text string. The underlying MS Excel worksheet storing the values and formulae is shown in Figure 7.

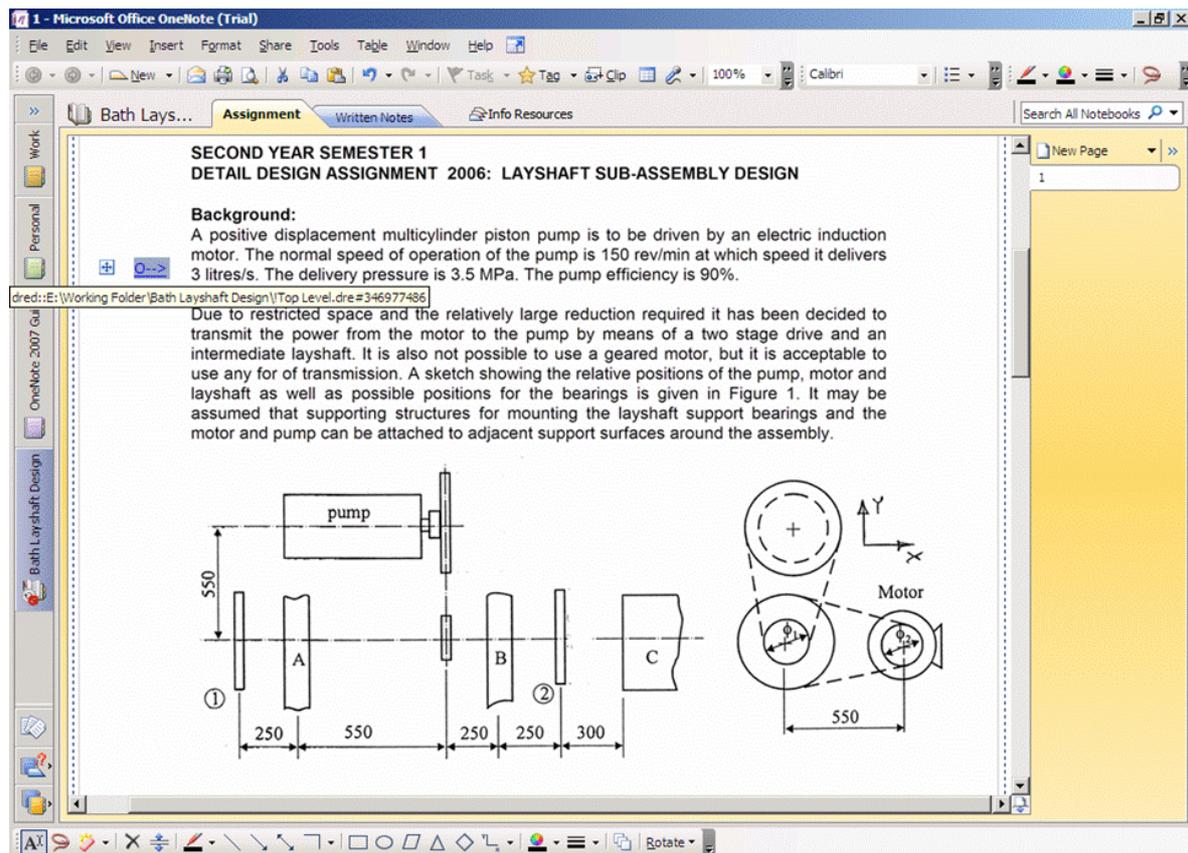


Figure 5. University of Bath Undergraduate Embodiment Design Exercise

The directed links show the dependencies in the worksheet formulae, which implement calculations that were originally done by hand. This style of use of DRed together with MS Excel, is intended to address the widely acknowledged problem of poor visibility of spreadsheet formula dependencies [15]; addressing this problem has recently been cited as the main selling point of Parametric Technology Corporation’s newly implemented ProENGINEER/Mathcad interoperability [16]. In this case the formulae, calculating, for example, the hydraulic power from the product of the pressure and flow rate, are so straightforward that there is no point documenting them explicitly in DRed. Thus, only the directed linkages between the dependent and independent variables are shown. Where formulae are more complex, they can easily be documented using equation objects residing in MS Excel, their images captured in DRed and linked to the objects representing their variables. Manual design tasks, such as a making a selection from a component handbook are signified by the Pert chart style *Task* elements, coloured green and crossed to show that the state of the task is completed, following the common Pert chart convention. Directed links lead from the elements representing input values to the selection task, and lead from the task to the element or elements capturing the output. For example, the “Choose belt length” task requires as inputs the symbolic belt type description 8MGT, and the number of grooves in each pulley, 36 and 90 respectively. All these values are unit-less, signified by the units specification *ul*. The output of the task is the choice of a

standard length belt giving a centre distance of 543.6mm with these pulleys. Also linked to this *Task* element is a *File* element with a MS OneNote η icon which is a bidirectional hyperlink to the location on the scanned page of the belt drive manual used for this selection.

Linked to the motor/lay shaft centre distance element described above, are two elements where the left hand numerical cell has a solid green background. These are examples of how numerical design constraints may be represented and maintained using this approach. They represent the constraint in the design specification that the motor to lay shaft centre distance must be 550 ± 50 mm. This is represented by two formulae calculating the distance of the value from its limit, so that when the result of the formula becomes negative the constraint is being broken. This is clearly shown by using cell conditional formatting to switch the background colour from green and red if the formula value is less than zero.

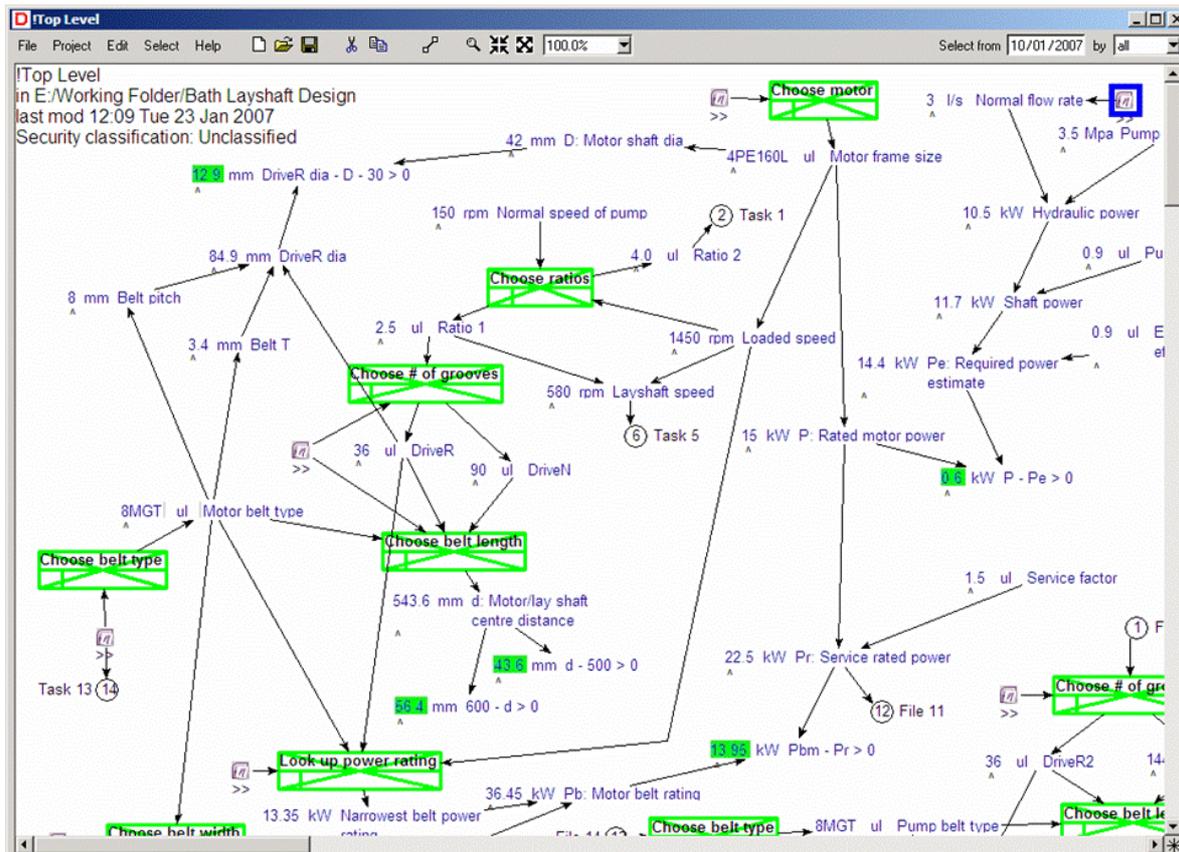


Figure 6. MS Excel-linked DRed chart capturing lay-shaft transmission design activity

The format of the underlying MS Excel file shown in Figure 7 has been carefully chosen, so that as well as allowing the capture of clearly understandable element images into DRed, it is compatible with typical parametric solid CAD systems. Most PC-based CAD software can create Object Linking and Embedding (OLE) links from multiple part and assembly files to one or more MS Excel worksheets, so that parameters and their values defined and calculated in the worksheet can directly drive the geometry, with automatic updating of parts if the parameter value in the worksheet changes. In the background, Figure 7 shows the result of linking the example MS Excel spreadsheet in this way to an Autodesk Inventor part file. It can be seen that the parameter names and values defined and calculated at the pre-CAD stage have been parsed successfully and are available to size the geometric elements that will later be created. Note column A in the worksheet, which is not part of the adjoining cell selections linked into DRed. This column is the source for the symbolic names of the parameters imported into the CAD tool. It is required by Inventor that every cell in that column must contain a unique symbol, otherwise the parsing process will terminate at that row. This condition has been maintained by inserting the unique DRed-generated bookmark names of the adjoining cells into this column as default values. For rows representing parameters that are being used in the CAD model of the design, it is clearly preferable that the parameter names should be more meaningful, and thus the

further extensions and refinements to the software, aiming to improve the ease of use for designers and to maximise the benefit. For example, it would be straightforward for DRed to detect when one or more input values to a task had changed in the underlying worksheet, and automatically mark the task as needing to be redone by changing its status from completed to pending. Thus, where a design needs iteration or redoing in a slightly different context, capturing the design in this way might allow the calculations to be redone automatically and the software to guide the user as to which manual tasks need redoing, in what order, with what inputs.

The implementation of these software developments will, as was done in the early stages of DRed, adopt a rapid prototyping approach, placing the new facilities into the hands of users as early as possible so that their feedback can guide the development. At that stage the benefits or otherwise will be easier to assess accurately by use in practice, and to quantify by experiment.

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 Department of Trade and Industry through the HIPARSYS project and Rolls-Royce plc through the UTP for Design. The authors are gratefully indebted to Lian Ding, Matt Giess and Goh Yee Mey of the University of Bath, for conducting and preparing the materials for the lay shaft design exercise case study.

REFERENCES

- [1] Kunz, W. and Rittel, H.W.J. Issues as Elements of Information Systems. *Center for Planning and Development Research*, Berkeley, USA, 1970, Working Paper 131.
- [2] Conklin, J. and Begeman, M.L. gIBIS: a hypertext tool for exploratory policy discussion. *ACM Transactions on Information Systems*, 1988, 6(4), 303-31.
- [3] Bracewell, R.H. and Wallace, K.M. A Tool for Capturing Design Rationale. In Folkesson, A., Gralen, K., Norell, M. and Sellgren, U. (Eds.) *Proceedings of the 14th International Conference on Engineering Design*, Stockholm, Sweden, 2003, 185-186.
- [4] 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. In Dutoit, A.H., McCall, R., Mistrik, I., and Paech, B. (Eds.) *Rationale Management in Software Engineering*, Springer 2006.
- [5] 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.
- [6] Longueville, B. KINOA: a collaborative annotation tool for engineering teams. In *International Workshop on Annotation for Collaboration*, Paris, November 2005.
- [7] Cabanac, G., Chevalier, M., Chrisment, C. and Julien, C. A Social Validation of Collaborative Annotations on Digital Documents. In *International Workshop on Annotation for Collaboration* Paris, November, 2005.
- [8] Fogli, D., Fresta, G., Mussio, P., Marcante, A. and Padula, M. Annotation in cooperative work: from paper-based to the web one. In *International Workshop on Annotation for Collaboration*, Paris, November, 2005.
- [9] Overview of the Microsoft ActiveX Hyperlinks API. Available from <http://msdn.microsoft.com/workshop/misc/hlink/overview/overview.asp> [January 2007]
- [10] Sidewalk: Superimposed Information for Digital Libraries in an Educational Environment. Available from <http://datalab.cs.pdx.edu/sidewalk> [January 2007]
- [11] Murthy, S., Maier, D., Delcambre, L. and Bowers, S. Putting Integrated Information in Context: Superimposing Conceptual Models with SPARCE. In: *Proceedings of the First Asia-Pacific Conference on Conceptual Modelling*; Dunedin, New Zealand, January 2004, 71-80.
- [12] Murthy, U., Richardson, R., Fox, E.A. and Delcambre, L. Enhancing Concept Mapping Tools Below and Above to Facilitate the Use of Superimposed Information. In *Concept Maps: Theory, Methodology, Technology: Proceedings of the Second International Conference on Concept Mapping*, San José, Costa Rica, 2006.

- [13] Universal Resource Identifiers. Available from http://www.w3.org/Addressing/URL/URI_Overview.html [January 2007].
- [14] A Tcl interface to ImageMagick & GraphicsMagick. Available from <http://tclmagick.sourceforge.net/TclMagick.html> [January 2007].
- [15] Whitley, K.N. Visual Programming Languages and the Empirical Evidence for and Against. *Journal of Visual Languages and Computing*, 1997, 8(1), 109-142.
- [16] Fireman, J. Pro/ENGINEER and Mathcad - Joining the Puzzle Pieces. Available from http://www.imakenews.com/ptcexpress/e_article000665925.cfm?x=b11.0.w [January 2007].

Contact: Dr Rob Bracewell
Engineering Design Centre
University of Cambridge
Department of Engineering
Trumpington Street
Cambridge CB2 1PZ
UK
Phone +44 (0)1223 760559
Fax +44 (0)1223 760567
e-mail rhb24@cam.ac.uk
URL <http://www-edc.eng.cam.ac.uk>