INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

VISUALIZATION OF INFORMATION TRACEABILITY IN PRODUCT DEVELOPMENT

T. Martinec and N. Pavković

Keywords: traceability, visualization, diagram, network

1. Introduction

This paper reports on usage of computer-based diagramming tools towards establishing and visualizing traceability of engineering information in product development processes. The authors combined and merged existing computer tools and methods for creating diagrams with the proposal of newly developed prototype tool and methodology for visualisation of relationships (traces) between design information objects. Presented research has been conducted in collaboration with one of the development departments of large company that produces all kinds of equipment for power plants. This department develops and produces control systems for general industrial purposes and especially for power plants and railway vehicles.
The research started with the thorough study of the technical documentation and design process structure for several already finished major projects. After that the authors were immersed in an ongoing development project (which lasted approximately for one year) as "proactive observers" with following main goals:

- to analyse the needs and possibilities of establishing and visualizing traceability in project documentation;
- to propose and provide methodologies of using existing diagramming tools;
- based on findings from project participation, to propose and develop a prototype tool and methodology that will facilitate the creation and visualization of traceability links.

Diagrams augment cognition [Salustri et al. 2008]. As such, a good diagram augments the capacity of the diagram's user to achieve goals. Visualization literally "makes visible" (or "evident") things that might not otherwise be so [Salustri et al. 2008].
Following such an approach we wanted to answer following research questions:

1. How can diagrams be used as a means of information recording in order to establish traceability in product design?
2. In what way could created diagrams (e.g. IBIS diagrams) be integrated into project documentation?
3. What other types of information could be attached to project documentation in order to create context around it and establish traceability?
4. How to display enriched project documentation and visualize traceability links?

In the first phase of observing (early design phases) we started with the analysis of design communication issues (both in design team and between team and external partners). This phase resulted in proposal of methodology for capturing the "communication timeline" specific for observed design environment (elaborated in chapter 2). People communicate and exchange engineering information across organizational and discipline boundaries, so they reuse existing information in new and unpredictable contexts. Because of the lack of the formal representations of the complex engineering design information, these information exchanges still partly occur informally. Capturing, organizing and visualization of some essential team communication data could improve retrieval of the engineering design information objects as well as correct interpretation of their content [Pavković et al. 2013]. The next research phase has been focused to design rationale capturing with IBIS based tool with a goal to link design rationale with product architecture visualization. In the third phase of research (while the observed project was in the final phase) the primary need of industrial partner was to establish the full traceability of main functional requirements. This led us to development of a tool and methodology which comprises and integrates previously proposed diagrams in a "diagram network". Such a network includes interrelated diagrams as well as environment for establishing and visualizing links between all kinds of files that are parts of project documentation.

1.1 Related work

Design rationale may be viewed as traceability of design thinking and the decision process. Shipman and McCall [1997] view design rationale as a topic that implies different things to different people, some describing it as the capture and potential reuse of normal communication about design. They proposed an integrated approach to design rationale where design communication is captured and, over time, incrementally structured into argumentation and other formalisms to enable the improved retrieval and use of this information. There are many similarities and overlapping issues between traceability issues and design rationale capturing.

Aurisicchio and Bracewell [2009] proposed a novel approach to designing and its documentation by integrated diagrams formalised into a templated structure and illustrated by means of a case study in the aerospace engineering industry. This line of research continued in Eng et al. [2012] where authors emphasise a need for increased investment in flexible visual tools to aid human thinking. Dai et al. [2012] used an IBIS based tool for the analysis of non-functional requirements.

Salustri et al. [2008] made a review of existing diagramming tools and they concluded that:

- Simplicity is important. The simpler the tool – even though its scope may be limited as a result – the easier it is to use, and the more likely users are to adopt it willingly and "naturally".
- Network hypergraphs are essential. The richly interrelated information elements typical in early designing are highly coupled, and representing those relationships is essential.
- Diagram layout is essential. A proper layout for a diagram can actually simplify it without loss of semantics.

Based on their findings the authors argue that there is no existent tool fully suitable to engineering design support purposes and that a new framework for diagramming tools must be developed.

2. Information traceability through diagrams

This chapter describes our research work on establishing engineering information traceability using diagram tools as means of information recording. Presented approach mainly relies on findings from our previous research project on traceability of engineering information [Marjanović et al. 2011], [Štorga et al. 2011]. This project comprised a preliminary traceability case study conducted with the same industrial partner as the work described in this paper [Pavković et al. 2012].

Information displayed in diagrams is structured using a concept of nodes and links between the nodes - making them information containers. A wide range of data can be embedded, including digital entities storage information, such as hyperlinks to computer-stored files. There is no limit in terms of file types that can be linked (CAD, spreadsheets, text documents...), including other diagrams. Creation of links between diagram files forms a diagram network. Such network allows users to cross boundaries of a single record and browse information spread in multiple design episodes. Embedding diagram hyperlinks into nodes is the key of establishing diagram networks and allowing users to easily shift from one record to another.

We argue that diagrams are convenient for both fast recording and retrieving of particular tracing context on design episode level, and consider diagram networks as the basis of well-established traceability on project level. A computer-based diagramming tool was used to test the methodology. It features basic node-link creation, formatting and arrangement, predefined IBIS nodes, image import, hyperlink embedding, ontology support and search mechanisms. Basic nodes are composed of geometric shapes, labels, notes and keywords, while IBIS nodes also include an icon depending on the type (issue, answer, argument...) and status (accepted, likely, rejected...).

2.1 Proposed diagrams

Several types of diagrams were introduced throughout the methodology and diagramming tool implementation on the ongoing project. These diagrams cover communication visualization, product structure and specification, and design rationale.

2.1.1 Communication timeline

The main idea of the "communication timeline" diagram was to create and regularly update a general project timeline, specific for the observed design environment, as a starting point for all future diagrams. This timeline follows the entire product development process, from consolidation of project and product requirements in early development phase, to verification and validation tests in late development phase, and covers essential development events and team communication data. Timeline nodes are e-mail messages, meeting notes, issues and discussions, arranged and grouped linearly as they appear through time, thus representing project view from the perspective of a development team leader. Figure 1 shows linear communication timeline, divided into multiple development phases. Each node can contain hyperlinks to other created diagrams, described hereafter.

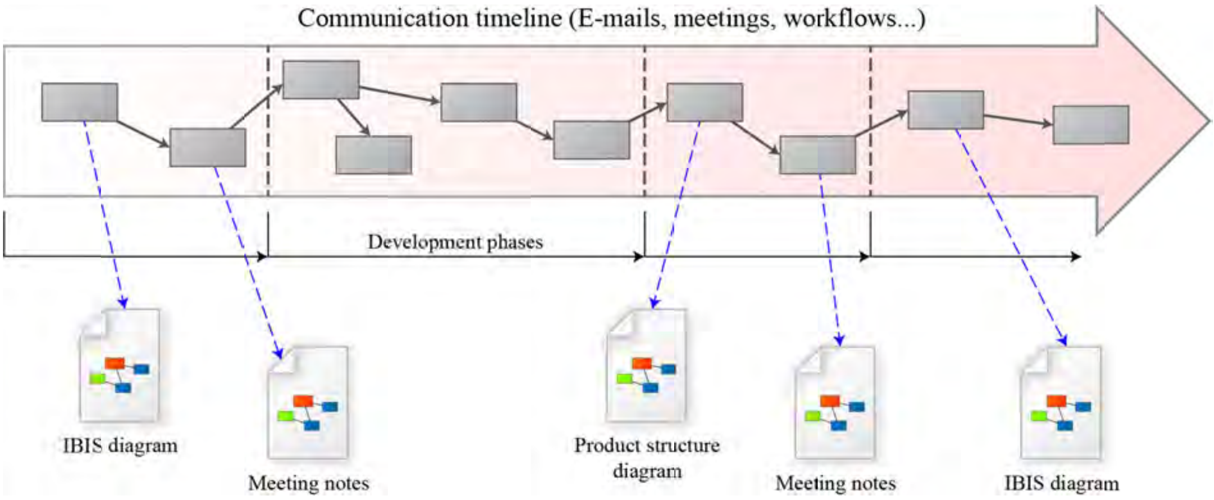

Figure 1. Communication timeline scheme

The capture of design information through linear text documents is a challenge because of the complex and fast-flowing nature of design activities [Aurisicchio and Bracewell 2013]. This also applies to meeting notes, so far written as text documents. Meetings are dynamic, non-linear, thus cannot be adequately replicated using only text documents. Such notes are vast, poorly structured and hard to trace or search. Replacing text documents with diagrams proved that they are more convenient means of meeting information recording. Meeting diagrams are clear, well structured, easy to follow, and linkable which makes them a part of the traceable diagram network.

2.1.2 System architecture and product specification

Working on power electronics and control concept design includes the establishment of system architecture and product specification. System architecture can be defined as a structure composed of components, and rules characterizing the interaction of these components [Jones 1993]. This definition can apply to both software and hardware system architecture. Just like functional breakdown structure, system architecture is usually represented using diagram drawings. By mapping these diagrams with the tested diagramming tool, we were able to link product description elements with communication timeline and design rationale, thus spreading overall diagram network. System architecture diagram is a starting point of product description visualization – a product structure overview. Once created, each node representing a component or a module, and each link representing their interaction, can contain hyperlinks to new, detailed description diagrams, e.g. a module specification diagram, as shown on Figure 2.

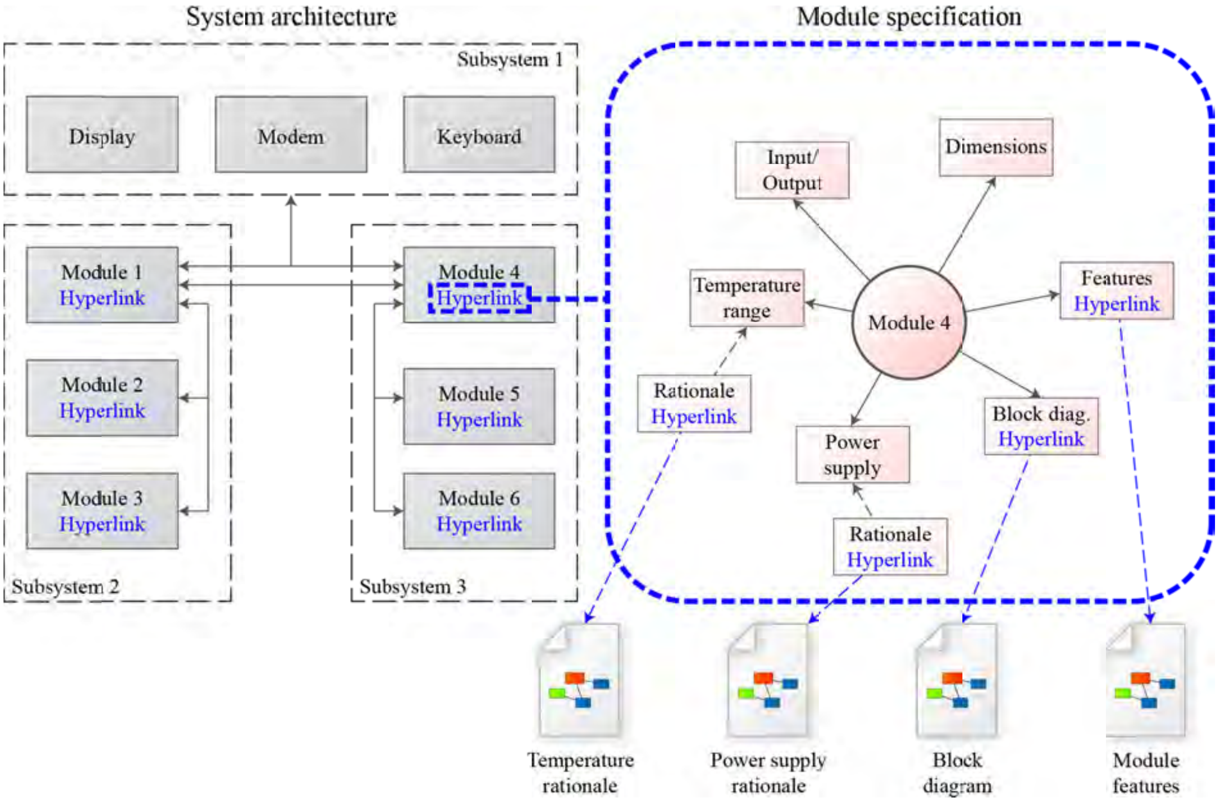

Figure 2. An example of system architecture and module specification diagrams

Product specification diagrams give an extensive description of product components, modules, subsystems, assemblies, etc. Concept design phase requires most of the product specification to be determined, usually in the form of a concept design text document, as a list of features and parameters. This is, again, not convenient for recording the decision making process. Assigning discussions, comments and design rationale to a text document makes it vast and poorly structured. Transforming lists into diagrams makes specification easier to organize, update and more important, it makes it linkable with other diagrams, such as decision making diagrams. This way every important decision or parameter value becomes a part of the growing traceable network.
In order to assure faster and simpler diagram generation, templates were prepared. Templates consist of default specification categories, which appear in most projects within the department. Blank nodes are arranged in a proper layout and prepared for parameter entry and hyperlink embedding.

2.1.3 Design rationale

The purpose of building the diagram network is the ability to track every important piece of data or information back to its source. These sources are often supported with rationale, which furthermore is often not adequately recorded, partly due to lack of time and partly due to tools used. All so far described diagrams serve as an infrastructure or pathways for users to access requested information. It was logical to continue recording design rationale information using diagrams.
When it comes to diagrammatic recording of design rationale, we decided to use IBIS diagrams (Figure 3). The Issue Based Information System (IBIS) consists of a tree or directed graph, where nodes representing issues to be resolved, alternative solutions, and arguments in favour and against, are linked by arcs [Kunz and Rittel 1970]. IBIS diagramming tools have already proven to be very effective and accepted among the industry, due to their simplicity and universality. Tested tool provides predefined IBIS nodes of various types and statuses, which makes design rationale easy to

capture, whether simple or more complex problems are being solved. The tool was especially handy for the recording of meeting discussions and problem solving, making design rationale easier to record compared to meeting notes writing. Also, retrieving of recorded information from diagrams is more efficient due to branched structure, rather than reading linear text documents. This way, users are directed to retrieve only information they need, rather than processing all of the written rationale.

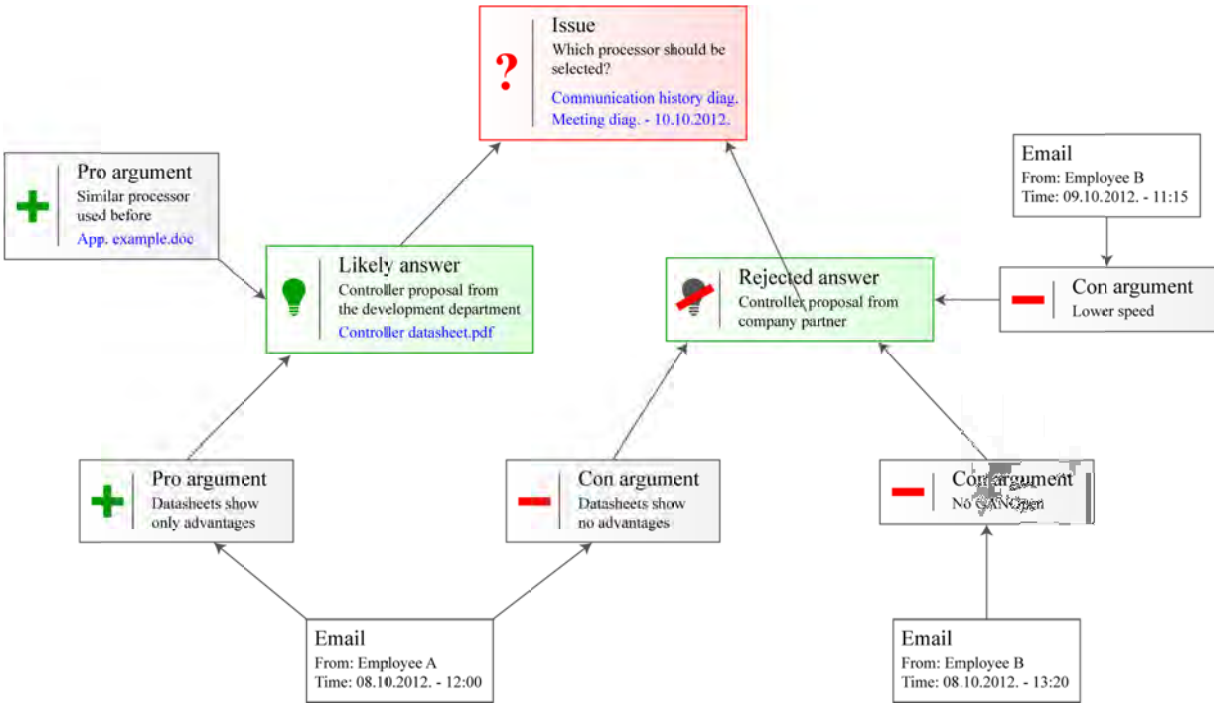

Figure 3. An example of an IBIS diagram

2.2 Diagram integration into project documentation

One of the benefits of using diagram records is the ability to embed hyperlinks to external files, such as project documents, into diagram nodes. This, however, doesn't integrate diagrams into overall project documentation. In other words, it doesn't make project documentation a proper part of the diagram network. This is because links between diagram nodes and external files are unidirectional, meaning they only lead from diagram nodes to files, but not from files to diagram nodes. This problem makes diagram network hard to integrate into project documentation. Figure 4 explains the problem of unidirectional links. For example, there can be a discussion which component solution to use, with multiple alternatives proposed. Each alternative is supported with arguments and hyperlinks to external files that describe the solution. The question is, how and where to store this decision making diagram?

Without recurrent hyperlink from project documentation to diagram files it is hard for users to know if some project CAD or datasheet file was considered an alternative in the decision making process, and thus has instances in the diagram network. The only way users can check this is to browse the entire diagram network in order to find these instances. This is in no way efficient or acceptable. Multiple options have been taken into consideration to assure bidirectional links.

First option is to save diagram files into directories of external files they refer to. This is, however, impracticable for several reasons. If some directory has multiple files, it is hard to discern which of them is linked to the diagram. The problem gets worse when a directory contains multiple diagrams. There is also a problem of a single diagram referring to multiple files that are not located in the same directories. This means that the same diagram would have to be saved to multiple directories, thus becoming very hard to update and link with the rest of the network – at least without using PDM or PLM software support. Ultimately, this option doesn’t provide actual bidirectional links, it only relies on users ability to recognize that there are unidirectional hyperlinks from diagrams to certain files.

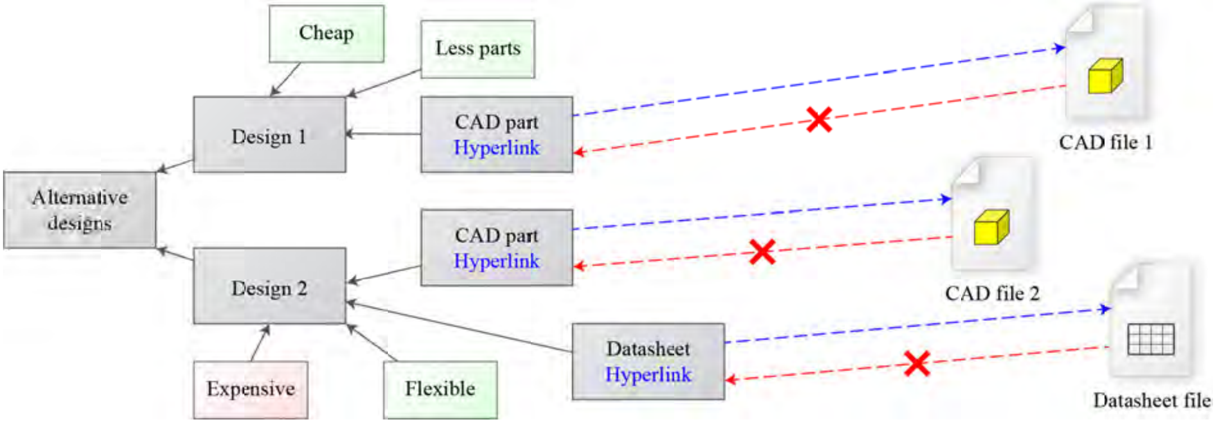

Figure 4. Unidirectional hyperlinks between diagram nodes and external files

Second option is to enable hyperlink embedding in software used to open various file types. For example, a text file being linked from diagram should have an opposite direction hyperlink when opened in a default text editing software. Every used software should be additionally coded to enable hyperlink embedding. There has been some work on this approach [Aurisicchio and Bracewell 2013] and a tool was developed to support bidirectional hyperlinking to and from bookmarked ranges in a variety of external document types, such as MS Word, MS Excel and others. This approach, however, requires a lot of work on software modification, especially when large number of different applications are being used through the product life cycle. On the other hand, some of the software don't even allow this kind of modification, meaning bidirectional hyperlinking cannot be established for all file types.

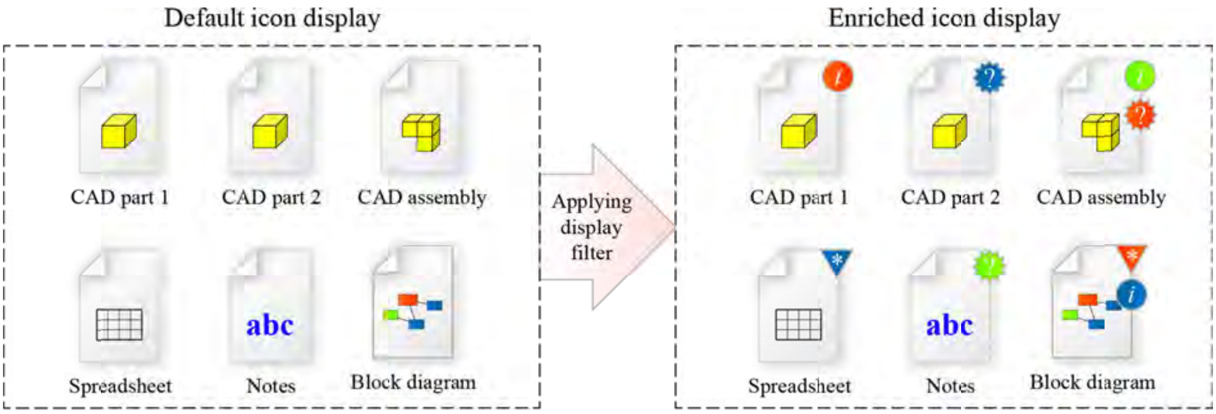

Figure 5. Enrichment of file icon display

Rather than modifying a wide range of software applications, we decided to find another way of hyperlink embedding. The goal was to minimally change users' computer usage habits and not to introduce just a new set of time-consuming software tools. One step before opening a project file is browsing for it among computer or server content. Without using PDM or PLM software this usually goes through the interface of Windows Explorer, which visualizes directory structure and file types. Users locate files by their name, icon and description. Modifying these properties allows us to enrich computer-stored files, more precisely project documentation, with additional information, such as traceability links, statuses, etc. Figure 5 shows the concept of modifying file icon display by changing visual representation of files in Windows Explorer, as some sort of display filter was applied. Similar concept can be applied within detailed view, by changing or adding textual properties to the file.

Changing file representation in Windows Explorer is the first step to bidirectional links between files and diagrams, files and other files, or even files and directories. If some project document is linked to a diagram, its representation can inform the user that such link exists. By selecting the document user can retrieve additional information and open the diagram file through the given hyperlink. This way, an actual bidirectional hyperlink exists between diagrams and external files.

The approach of modifying file display requires only the development of a browsing interface, similar to Windows Explorer, which allows users to browse project content and enrich it with a wide range of information, including hyperlinks.

3. Project explorer environment

The development of the project explorer environment was started mainly to integrate diagrams into project documentation, but the application was further upgraded with other useful features and is still in development phase. Two main objectives were set at the start of the development:

- Allow users to manually link diagrams with computer-stored files and display these links in the explorer interface
- Facilitate diagram creation with templates since the tested diagramming tool doesn't support template importing

New development objectives were additionally set, including:

- File to file (or directory) linking, using the same principle as in diagram to file linking
- File enrichment using ontology
- File status association and status display in the explorer interface
- Automatic visualization of created links in an interactive diagram form

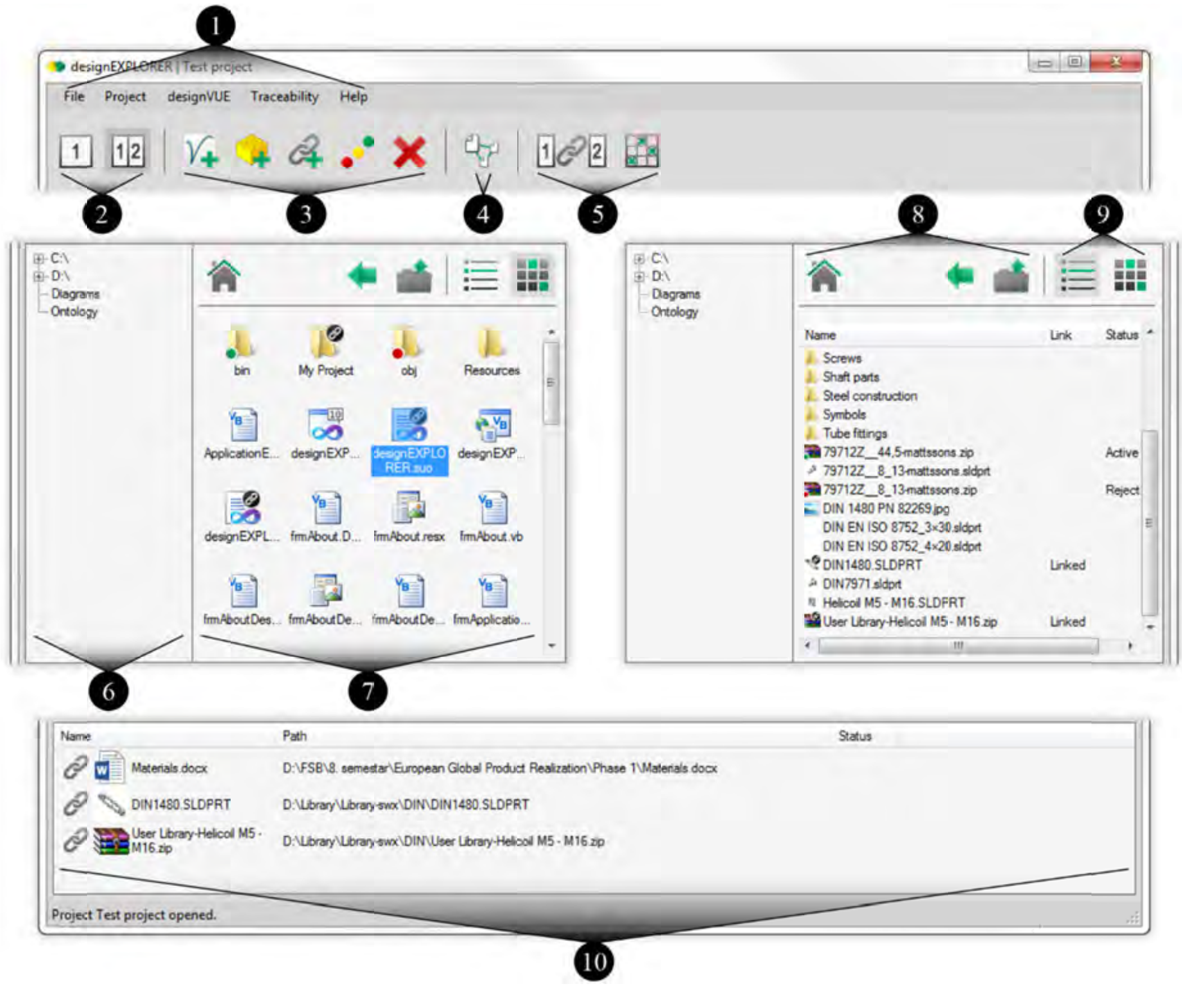

Figure 6. Main interface of the project explorer application

The application is conceived as a central tool for the creation of diagram networks. The diagramming tool, now a part of the environment, is supported with automated diagram storage and template selection. File browsers enable navigation through computer (server) content, and thus serve as Windows Explorer substitute. Files that are displayed in browsers can be manually linked to other

files, directories, diagrams or ontology, and associated with statuses. Links can furthermore be visualized either manually by exporting node-link files, or automatically with the developed diagram network visualization tool.
Main application features can best be described through the main interface overview shown on Figure 6, where numbers indicate elements with following functions:

1. Menu bar contains mainly project functions. Project can be defined as a record of all link, diagram, ontology and status data. Every project is given a name, and once created it can be saved, opened and edited. Also, each project can use a different subset of ontology, prepared to be linked with files. A single diagram network and its boundaries are defined with a single project file.
2. Two browser view buttons switch the additional browser on or off. Additional (second) file browser facilitates link creation between two selected files, especially when one of the files is a diagram or an ontology element.
3. Linking tools include diagram, ontology and other files linking buttons, status association button and a button for link deletion. Diagrams can be created either from scratch, or through a template, using the integrated diagramming tool.
4. Traceability link visualization button draws a diagram of all files that are in any way linked with the selected file.
5. Links between files displayed in two browsers can be created in two ways. One is to select a single file in both left and right browser and use the link selection button, and the other is through a content matrix, in which rows represent left browser content, columns represent right browser content and matrix fields represent links between them. These fields can be empty, meaning there is no link between corresponding files, or filled, meaning the link exists.
6. Directory tree panels display computer directory structure including diagrams and ontology directories, for faster browser navigation.
7. List view panels display the content of browsers current directory. Files can be opened on mouse double-click.
8. Navigation buttons change browsers current directory into home, back or one level up. Home directories of both folders are defined within project creation.
9. Icon view buttons change browsers list view between large icon and detail view.
10. Link browser displays all files that are linked to currently selected file in the left or in the right browser.

3.1 Traceability links between files

All link data is stored inside project files, so computer content is not in any way affected or modified. The only things needed to establish the link between two files are their storage paths. When a file is associated with a traceability link it becomes a part of the "link matrix".

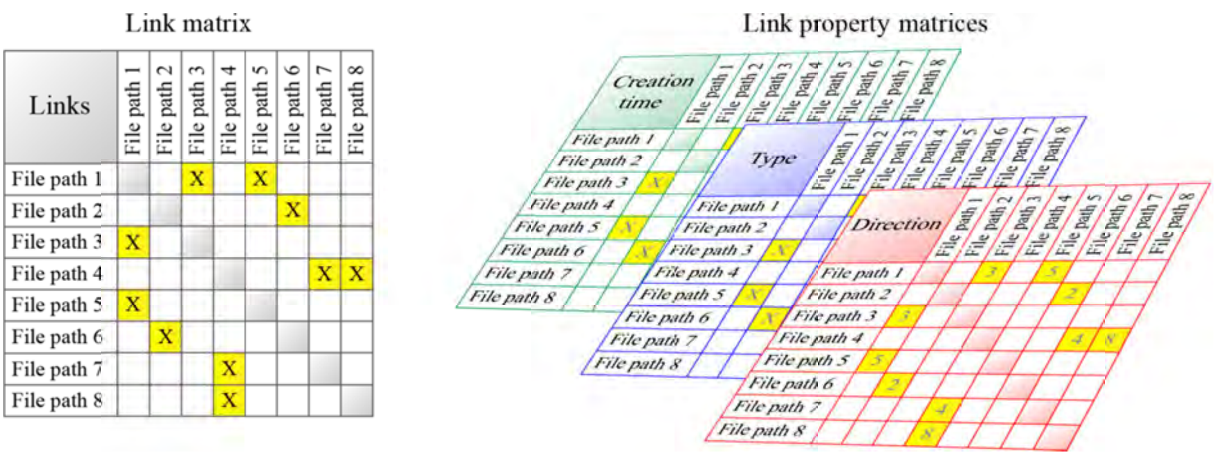

Figure 7. Link matrices scheme

Link matrix is a symmetrical matrix, whose rows and columns represent file paths which are associated with one or more traceability links. If the link matrix field is filled, a link exists between corresponding row and column file paths, as shown on Figure 7. Link matrix is convenient for fast code procedures, e.g. search loops.
In order to be simple, the main link matrix contains only data about the existence of links between files. If there is a need for link properties recording (such as link direction, link type, creation time, etc.), a new set of "link property matrices" can be created, as shown on Figure 7. These matrices have the same structure of the main link matrix, but their fields represent property values. New type of property can be easily added by creating new property matrix. When a traceability link is located in the link matrix, its properties can be found on the same address of the property matrices, thus reducing search loops volume and processing time.

3.2 Automated traceability link visualization

The prototype tool can visualize information in two ways, on two different levels. The first is on the browser content level, as shown on Figure 8, where file icons are automatically modified depending on whether the files are linked or associated with a status.

Figure 8. Different types of file icon display in the explorer tool

Other way of information visualization is on the diagram network level, by visualizing all established traceability links. An interactive diagramming tool was developed to automatically visualize diagram networks for the file selected in the explorer. Each file, diagram, ontology element or directory that is in any way linked with the selected file is represented in the form of a diagram node. Traceability links between files are represented as diagram links. The interface of the developed diagramming tool is shown on Figure 9. Nodes and traceability links are automatically placed on the diagram surface, which can be navigated using "pan and zoom" tools. Users can change node position manually in order to change diagram network layout.

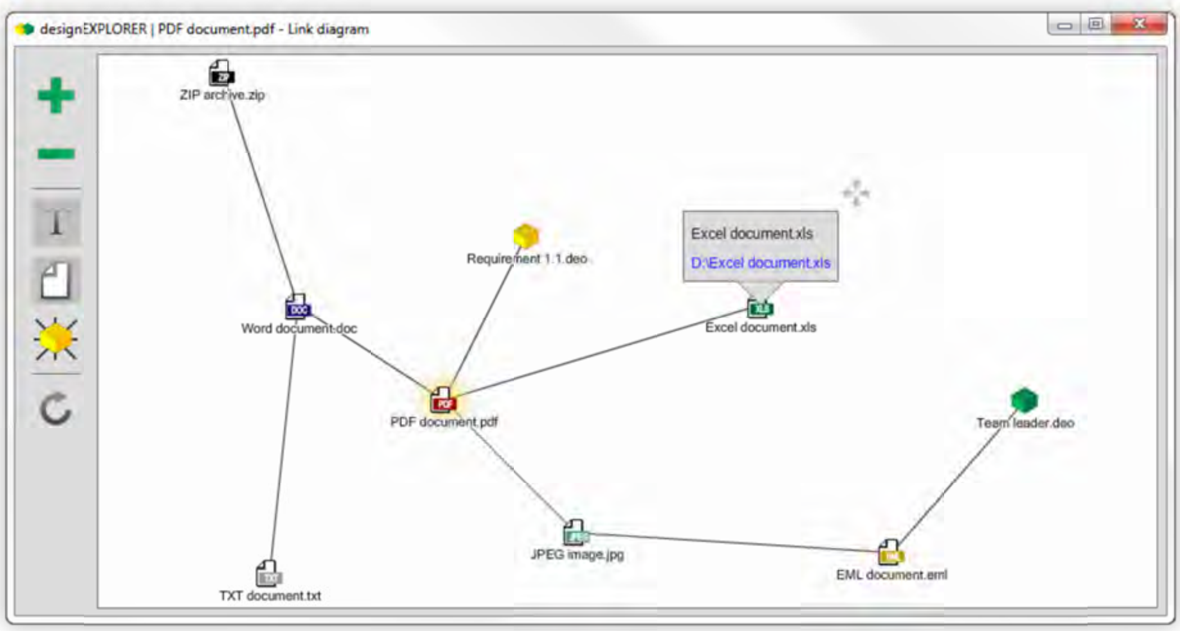

Figure 9. An example of automated diagram network visualization

Nodes can be displayed as file icons or as black-spot nodes, and with or without file name labels. Icons make files easier to distinguish and are convenient for simple, brief diagrams. Spot nodes are used for more complex diagrams, with a dense node network. Hovering the mouse over a node highlights only adjacent nodes by fading all nodes that are not directly linked with the selected node. Clicking a node makes an "info box" pop out, containing additional information about the file. Additional information includes a hyperlink to the file path, which triggers file opening in the default application.

4. Implementation and requirement traceability

First phases of the ongoing project were recorded using only existing diagramming tools, whose implementation resulted with several diagram types and methodologies. Recorded diagrams initiated the creation of the diagram network and established information traceability in early design phases. However, they were difficult to integrate with the rest of the product documentation and thus unsuitable for everyday use and long-term implementation. The new prototype tool for diagram integration and traceability link visualization was ready to implement in later (final) design phases, primary to facilitate hardware and software testing processes. It provides a complete diagram network solution in a form of a single application environment, in order to avoid software accumulation and greater change of computer usage habits.

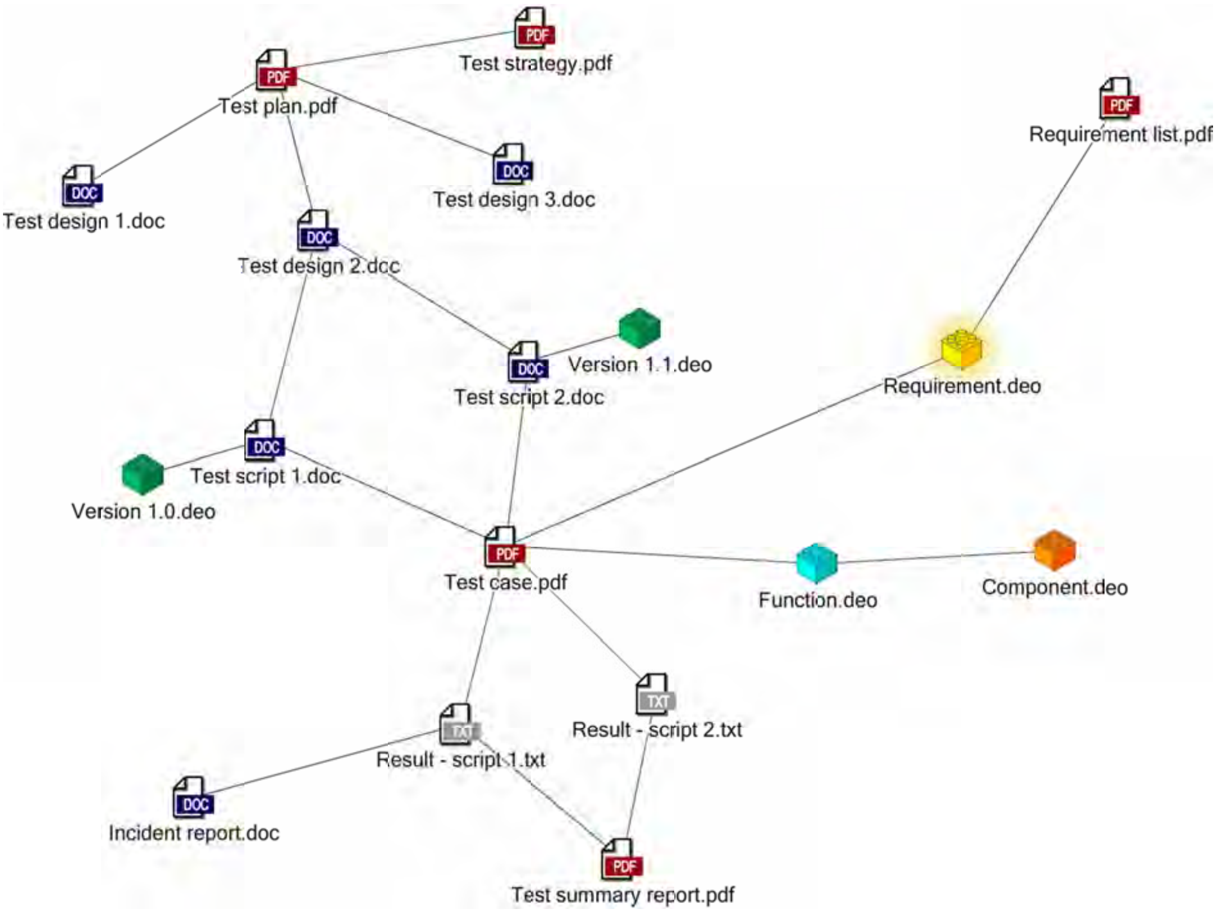

Figure 10. Requirement and verification traceability visualization

Industrial partner's primary need in final design phases was to establish full traceability of main functional requirements since they are tested through strict processes of verification (whether the product meets design requirements) and validation (whether the product meets the purpose requested by customer). Previous collaboration showed that these tests generate a huge set of relatively small documents in which many authors from several different departments (employees) participate.

Documents are highly interrelated from various viewpoints – hierarchically, by evolution of content and by referring to content fragments [Pavković et al. 2012]. Recording these relations and representing them in a diagram form would largely facilitate product verification and validation. It is important to visualize the entire traceability chain: from requirement sources, through requirement definition lists and component functions, up to test results. An example of such chain visualization, made with the tested prototype tool, is shown on Figure 10.
Furthermore, requirement traceability chains can be supported with ontology and with the rest of the diagram network, for better context understanding since the testing process is also very interesting from the viewpoint of utilization, especially from the viewpoint of knowledge reuse. Testing results, especially test incident reports are very interesting candidates to be back-tracked in utilization process [Pavković et al. 2012].

5. Conclusion and further work

Diagrams covering communication visualization, product structure and specification, and design rationale were introduced throughout the research and diagramming tool implementation on the ongoing project. By creating links between diagram files a diagram network was formed. Such network allows users to browse information spread in multiple design episodes. The purpose of building the diagram network is the ability to track every important piece of data or information back to its source. All described diagrams serve as an infrastructure or pathways for users to access requested information.
Difficulties of diagram integration into project documentation were solved with a newly developed explorer environment. Users locate project files (using the explorer interface) by their name, icon and description. Modifying file representation properties allows users to enrich stored files, more precisely project documentation, with additional information such as traceability links to relevant diagram files.
Files that are displayed in explorer browsers can be associated with statuses and manually linked to other files, directories, diagrams or ontology. Links can furthermore be visualized with the developed diagram network visualization tool.
Prototype implementation showed that diagram network visualization facilitates requirement fulfilment verification, whereas only a quick diagram overview gives an insight of all requirements and the corresponding test documentation. This is particularly helpful when the verification is done by a third party.
There is still enough space for further development of proposed project explorer environment, with more ideas to come through additional application implementation on other projects. Some of the currently defined guidelines for the next development steps are:

- Implementation of search mechanism – The application currently lacks a searching mechanism, and its implementation would have a major role in traceability establishing, largely due to efficiency benefits. Search mechanism should allow users to search using keywords and result filters. Search results should be displayed as directories and files within the explorer interface. Search filters should enable users to limit the search to only specific file types, specific statuses, link number, file properties, etc. An example of search scenario is when a user gets a list of all requirements that are not yet verified or in any way linked to the test documentation.
- Improvement or replacement of current diagramming tool with a newly developed tool for manual diagram creation, as a part of the environment – Although the used diagramming tool proved to be very handy and effective, it has a lot of drawbacks when it comes to environment integration. If search mechanisms are to be added, diagram content created with the current diagramming tool would be out of the search scope. The other reason for the development of a new tool is its adjustment for the proposed diagrams and methodologies, including currently unavailable special node types.
- Assigning properties to traceability links between files – The link matrix is structured to be supported with additional property matrices. These matrices were not created for the prototype tool, but for the further development it is important to enrich traceability links with

information about their type, direction and creation time. Time records are often needed to achieve well established traceability.

- Windows explorer integration – Rather than being a standalone application, the environment tools could be a part of the upgraded Windows Explorer interface, making it even simpler to use and easier to accept among development teams.

References

Aurisicchio, M., Bracewell, R. H., "Engineering Design by Integrated diagrams", Proceedings of the ICED 09 Int. Conf. on Engineering Design, Stanford, USA, 2009.

Aurisicchio, M., Bracewell, R. H., "Capturing an integrated design information space with a diagram based approach", Journal of Engineering Design, Vol. 24, Issue 6, 2013, pp. 397-428.

Dai W., Aurisicchio, M., Armstrong, G., "An IBIS based approach for the analysis of functional requirements", ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2012), Chicago, USA, 2012.

Eng, N. L, Aurisicchio, M., Bracewell, R. H., Armstrong, G., "Mapping for design decision support in industry", ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2012), Chicago, USA, 2012.

Jones, A., "The Maturing of Software Architecture", Software Engineering Symposium, Software Engineering Institute, Pittsburgh, 1993.

Kunz, W., Rittel, H., "Issues as Elements of Information Systems", Technical Report Working Paper No. 131, Institute of Urban and Regional Development, University of California, Berkeley, 1970.

Marjanović, D., Štorga, M., Bojčetić, N., Pavković, N., Stanković, T., "EUREKA E4911 TRENIN project final report", www.trenin.org, 2011.

Pavković, N., Štorga, M., Bojčetić, N., Marjanović, D., "Facilitating Design Communication Through Engineering Information Traceability", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Volume 27, Special Issue 02, 2013, pp. 105-119.

Pavković, N., Tečec Ribarić, Z., Sviličić, T., "Traceability Case Study on Rail Vehicle Control Unit Development Project", Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, 2012, pp. 1567-1578.

Salustri, F. A., Bracewell, R. H., Eng, N. L., Weerasinghe, J. S., "Visualising early engineering design information with diagrams", Journal of Design Research, 6(1-2), 2007, pp. 190-217.

Salustri, F. A., Eng, N. L., Weerasinghe, J. S., "Visualizing Information in the Early Stages of Engineering Design", Computer-Aided Design & Applications, 5(1-4), 2008.

Shipman, F. M., McCall R. J., "Integrating different perspectives on design rationale: Supporting the emergence of design rationale from design communication", AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing 11, 1997, pp. 141–154.

Štorga, M., Marjanović, D., Savšek, T., "Reference model for traceability records implementation in engineering design environment", Proceedings of the ICED 11 International Conference on Engineering Design, Copenhagen, Vol. 6, 2011, pp. 173-182.

Tomislav Martinec, Research Assistant
University of Zagreb, Faculty of Mechanical Engineering And Naval Architecture
Department of Design, Chair of Design and Product Development
Ivana Lucica 5, 10000 Zagreb, Croatia
Telephone: +385 1 6168 515
Telefax: +385 1 6168 284
Email: tomislav.martinec@fsb.hr
URL: http://www.cadlab.fsb.hr