INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED15
27-30 JULY 2015, POLITECNICO DI MILANO, ITALY

MANAGEMENT AND VISUALIZATION OF RELATIONSHIPS BETWEEN ENGINEERING OBJECTS

Pavkovic, Neven (1); Martinec, Tomislav (1); Rohde, Danijel (2); Sikic, Bruno (1)
1: University of Zagreb, Croatia; 2: INETEC d.o.o., Zagreb, Croatia

Abstract

The paper discusses methods and interfaces for manual recording, management and visualization of relations between engineering objects aiming to suggest most convenient interface tools and methods for relating objects of a particular pair or set of domains. The tools/methods considered are browser, diagram and matrix. The browser interface was used to relate objects from hierarchically structured domains such as file system and product structure. Objects belonging to domain pairs without hierarchical structure were interrelated using the matrix interface. For engineering objects involved in multiple complex relations (both intra- and inter-domain) we argue that the most convenient technique is to create and visualize such a network of relations directly by drawing diagrams using the specific diagram methodology and tool. A matrix style interface concept integrates different methods and tools in a way that should contribute to building of shared understanding of design projects on the company level. Elements of interface could be considered as predefined contexts for relation recording and indexing as well as guidance for information and knowledge retrieval.

Keywords: Engineering objects, Relationships, Visualisation, Design management, Teamwork

Contact:
Prof. Neven Pavkovic
University of Zagreb
Faculty of Mechanical Engineering & Naval Architecture
Croatia
neven.pavkovic@fsb.hr

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

The increased complexity of product development process, especially in medium and large-scale projects, generates many situations with which existing design support tools and methods struggle to deal with. Design communication in large teams has to take place within the framework(s) of huge networks of complex dependencies (relationships) between engineering objects of various domains that are very difficult and expensive to be efficiently managed (Koehler et al., 2014). The aim of this paper is to analyse and compare the suitability and efficiency of several methods for managing and visualization of relationships between engineering objects.

Here we consider the *engineering object* (EO) as an instance of any kind of document, physical component or an "abstract" notion that arises or exists in product development process. For instance these may be: CAD part, calculation, detailed design parameter, requirement, function, partial solution, testing report, production specification, etc.

Presented analysis has been focused to two main issues: (1) improving the quality of design communication during the NPD process and (2) improving the understanding of past design projects when previous solutions and knowledge have to be reused.

The general network of relations between EOs is commonly (predominantly) being treated from the viewpoint of traceability. In this research we have applied a more general approach where created and visualized relations have other roles apart from traceability viewpoint (e.g. solution principle (concept) explanation, collections (sets) of related objects, design rationale capturing and mapping, etc.)

Based on research findings focused to traceability practice in industry (Marjanovic et al., 2011; Marcus et al., 2005) it is arguably obvious that it is impossible and unnecessary to establish and manage a “full network” of all existent relations between all engineering objects. A very large number of engineering objects belonging to many different domains exist in any sociotechnical system on levels of granularity that satisfy practical needs (Koehler et al., 2014; Koenigs, 2012). Therefore a compromise subset of relationship network must be extracted resulting with an acceptable cost/benefit ratio in terms of efforts spent on relation recording, management and visualization of relation network.

Such an approach leads to following research questions:

- *Which relations (between EOs from which domains) should be of primary concern (interest) for medium and large-scale design projects? (Concerning the communication improvement, shared understanding and knowledge reuse).*
- *Which method(s) of management and visualization are most suitable for each particular detected subset of beneficial relations?*

Such an approach implies the following classification of EO relations*: intra domain (relations between objects of same domain)*, between objects from *two different domains* and finally complex sets of relations between objects from several domains (*multi domain*).

By making information structures organized, modern visualisations provide means for user to interactively navigate and uncover the information engineers are looking for (Keller and Tergan, 2005). It is presumed that the user is often being unaware of the precise information location by which the information can be obtained or possesses incomplete specification relating the information necessary to perform search. Both of the latter could be the cases in the product development of the complex technical systems involving large data and information sets and multitude of stakeholders generating and interpreting information. In (Martinec and Pavkovic, 2014) we argue that diagrams are convenient for both fast recording and retrieving of particular tracing context on design episode level, and we consider diagram networks as the promising basis for EO relation network visualization on project level. A similar approach verified in design practice is proposed by Aurisicchio and Bracewell (2013). Visualized network of semantic relationships that exist within and across life cycle of engineering objects could help engineering designers to understand the existing information and reuse them in the right context. Research literature describes the impact of poor traceability practices on project efficiency. A decrease in system quality, increase in the number of changes, loss of knowledge due to turnover, erroneous decisions, misunderstandings, and miscommunication are some of the common problems that arise due to lack of or insufficient traceability of engineering information (Hurwitz and Kaufman, 2007; Pavković et al., 2013).

The application of the described research approach is illustrated through a detailed presentation of the case study where a new mechatronic device has been designed in a medium size company specialized for the development of devices for nuclear power plant examination and inspection. The design team has been asked to use and compare several different tools and methods for creation and visualization of relations between engineering objects.

2 METHODOLOGY AND RELATED WORK

Methods and tools used in conducted case study have been selected and developed based on findings from our several previously finished case studies (Pavkovic et al., 2012, 2011; Martinec and Pavkovic, 2014). This work is also grounded on approach of Aurisicchio and Bracewell (2013) and on complexity management methods developed by Lindemann et al. (2009) and Maurer (2007).

2.1 Characterizing relations and EOs

One of the starting points of the analysis that has to answer research questions is distinguishing and characterizing relations and EOs from the following points of view:

- Stability over time periods
- Levels of granularity
- The quantity of EOs in particular domain
- Differences between domains of EOs
- Cardinality of relations (binary or multiple)

From the dynamic point of view, product structure and/or product architecture (or at least their elements) could be considered as relatively static data structures (on higher levels of granularity) for majority of engineering design environments, e.g. the reuse of mechatronic systems in automotive industry. Although product structures of complex products could contain large sets of EOs and relations, they do not change significantly over project life-cycle (on higher levels of granularity). Therefore we assume that it could be cost-effective to build relation templates for such predominantly static structures. Such an approach could be considered as a semi-automated method, because engineers would reuse and update templates while generating sets of relations. In a similar manner we assume that the majority of relations between other EOs from different domains have a more dynamic character, but probably smaller sets of EOs will have to be related requiring less effort and time consumption. A comprehensive relation classification is presented in Spanoudakis and Zisman (2005).

2.2 Characterizing recording/visualization methods and interfaces

The tools for recording and visualization of relations that have been used during this research project could be divided into three basic interface concepts**:**

- **browser of hierarchical structure(s),**
- **graph (diagram) consisting various classes of nodes and edges**
- **matrix** (intra domain or two domain)

The main idea - creation (recording) of a particular (certain) relationship type between engineering objects - is the same for all three of these concepts.

The domain to which engineering objects belong in large scale determines the selection of best interface concept for recording and visualization of relations. Probably the simplest problem is the creation of relations in-between the content of the file system (when files are being treated as objects of one same domain). Using a browser interface, creation of relation network between files can result in a well-established traceability of project documentation. This is usually not sufficient to provide efficient and complete support to designers - management of complex engineering data requires creation of complex relationships between objects from multiple different domains. Therefore engineering objects should also represent abstract notions from various domains (requirements, functions, changes, design tasks), “physical” objects like elements of product structure (components), or employees. A step forward would be to use browser interface for relating (linking) files and other "non-file" objects represented as notions organized as taxonomy and/or ontology. Generally, we want to explore a methodology where a combination of several recording and visualization methods will work in synergy to serve as a framework for generation of network of relations in-between all domains mentioned above. The structure of this network should not depend on the type of the interface used to store or retrieve the data from it.

The developed **browser interface** consists of two "explorer windows" that display either the file system content or any other hierarchically structured data (Martinec and Pavkovic, 2014). This type of interface requires a minimal change in users' computer usage habits, since one uses explorer on a daily basis. The relations are being created by selecting explorer items and assigning a desirable relationship between them, similar to the use of source and target browser described in Marcus et al. (2005). After it is assigned, the relationship is emphasized through the change of icons of the related objects. Additional information about the objects and their relations can be retrieved by selecting the corresponding explorer item. Besides that, browser objects can be enriched with information such as statuses, comments or design rationale. Detailed explanation of the proposed browser interface concept can be found in Martinec and Pavkovic (2014).

Diagram interface enables the creation of nodes and edges that connect the nodes. Nodes represent the objects and edges represent the relations. Every created diagram, no matter which diagramming tool was used, is a record of objects and their relationships, and contributes to overall relation network on a project level. In contrast to browser which doesn't require learning of new tools, diagrams excel other interfaces in terms of augmenting cognition, as a good diagram augments the capacity of its user to achieve goals. Visualization literally "makes visible" (or "evident") things that might not otherwise be so (Salustri et al., 2008). Every diagram node is an information container, which can include information about digital entities storage, displayed 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. Adding links between diagram files creates a diagram network. Such a network allows users to cross boundaries of a single record and browse information spread in multiple design episodes. A similar approach is presented by Aurisicchio and Bracewell (2013).

Matrix-based approaches to complexity management are widely applied, so we decided to use them as the basic architecture of the relation creation (mapping) framework. Nowadays, the large variety of matrix-based methods in engineering can be classified by the quantity of the types of objects involved (Maurer, 2007). If relations within objects belonging to the same type (domain) are examined, the related matrices can be defined as intra-domain, e.g. commonly applied Dependency Structure Matrix (DSM), while matrices combining different objects belonging to different domains are referred to as inter-domain (Lindemann et al., 2009). Some applications make use of combinations of intra- and inter-domain matrices. Such an approach is called the Multiple-Domain Matrix (MDM) (Lindemann et al., 2009). Following this it is obvious that the matrix interfaces used to relate objects of the same domain are in essence intra-domain matrices. Additionally, interfaces used to relate objects of different domains are inter-domain matrices. If the overall network of relations on project level is to be represented using the matrix interface it would be displayed as a large MDM.

In our previous research projects we have developed prototypes of tools and interfaces that partially or completely support each of the above mentioned relation recording and visualization methods. Therefore we have decided to compare these methods based on sets of relations between selected pairs of domains. We wanted to examine which interface tool and/or method is the most convenient for creation and future management and visualization of relations between each particular pair of domains? "Convenience" is here primarily assessed from the viewpoint of cost/benefit ratio in everyday engineering practice, because manual creation of relations is difficult, error-prone, time consuming and complex (Spanoudakis and Zisman, 2005; Marjanovic et al., 2011a; Koehler et al., 2014).Therefore a compromise must be found which will provide satisfactory level of functionality (benefits) to engineers, but at the same time which will not require significant additional efforts to be developed, implemented and managed.

3 CASE STUDY - THE DEVELOPMENT OF MECHATRONIC PRODUCT

In cooperation with medium-sized, development focused company we have prepared and conducted a case study where we have been primarily focused to examine (explore) cost/benefit ratios for creation and recording various kinds of relations between EOs. Furthermore, we were interested how the proposed methodology that combines three proposed methods and interfaces might overcome problems and issues identified in our previous research.

Case study conducted in 2011 (Pavkovic et al., 2011) has been focused to visualization of indexing structures used to relate engineering design taxonomy and knowledge captured during design projects. Based on findings from this case study we argue that the following strategy is essential: When a need for information and/or knowledge reuse occurs, the interface and procedures for searching and/or tracing should rely on the same taxonomy/ontology visualization and navigation methods as it was for initial indexing. Such approach is important for better and easier understanding of search context and for providing interface efficient and user-friendly enough to be unobtrusive (as possible) to designer overloaded with tasks and complex software tools. However, taxonomy/ontology based indexing implies problems that may occur in process of initial definition of entities and hierarchical structure of elements (entities) for practical usage in certain environment. It is very difficult to propose one common taxonomy structure that will equally suit the needs of all participants in particular product development process. Which notions (entities) should be on the top level(s)? This positioning directly influences the amount of time user needs for indexing and searching processes. Various stakeholders in PD process have different focal interests, implying different views on taxonomy structure.

In the second previously conducted case study (Pavkovic et al., 2012) we have tracked a development process of complex mechatronic product - the vehicle control unit (VCU) for new generation of regional train. VCU is responsible for control, measuring, sequencing, protection, supervision and communication tasks in the whole vehicle. For this research we have selected an approach where identified product development process structure, documentation as well as product hardware and software components have all been mapped to subsets of notions from proposed ontology. Findings from this case study showed us that besides mapping EOs to ontology, it is even more important to establish relationships between engineering objects, as well as reference lists for object contents. In other words, focusing only on information (digital) objects traceability is not sufficient to provide a complete and appropriate support for design communication and/or information reuse.

Therefore the case study presented in this paper has been focused to visualization of relations between all kinds of engineering objects. We have monitored (tracked) the process of completely new design of mechatronic device for underwater inspection of welds. The project lasted for three months, the team has been composed of one experienced designer, one novice designer and one experienced IT specialist. They were asked to use a variety of previously described methods and tools to record and visualize relations between EOs. The design team extracted EO domains and sets of relations they considered to be of primary interest for design communication, for building a shared understanding of project on company level as well as for reuse in the future similar projects. Five EO domains have been dominant (Figure 1): digital documents (files), requirements, functions, components and people (form the viewpoint of responsibility). When there was a need identified to create relations between particular domains, all three interfaces were discussed and the most convenient one was selected and used. Here we must emphasize that the cooperating company does not use any kind of commercial PLM system, just internally developed web-based production data management system.

3.1 Matrix-style interface and examples of visualization methods

During the monitored project a basic concept of interface framework has been developed (Figure 1): a simple visualization of a matrix with multiple domains was chosen as the initial interface (upper part of Figure 1). It represents all the domains and relations between them considered during the project. Matrix cells on the initial interface represent a set of relations between at least one pair of domains. Diagonal cells represent relations that exist between objects of same domain, while the rest represent relations of objects belonging to different domains. Any particular design environment could build and adapt domains and relations in the initial matrix according to its own needs. Similar approaches to interface concepts could be found in (Beier et al., 2013) and (Wickel et al., 2013).

The main idea of the initial matrix interface is to provide entrance points to application of selected tools and methods for relation recording and visualization. These entrance points are embedded inside the cells of the corresponding row and column. Therefore, following the entrance point the user is guided to one of the recording and visualization interfaces (the one that was evaluated as the most convenient). For example, if a field representing relations in-between the documents domain is selected (mark A on Figure 1), a browser interface is opened, where relations are created by selecting documents through file explorer. Furthermore, some of the fields in the same way lead to matrix (mark B) and some others to diagram interfaces (marks C-F). If multiple diagrams or matrix records have been made for a particular domain pair, their references are all stored inside the corresponding field, and the user selects which record to open when information is being retrieved (e.g. multiple design rationale diagrams embedded inside the field named "functions that imply design problems").

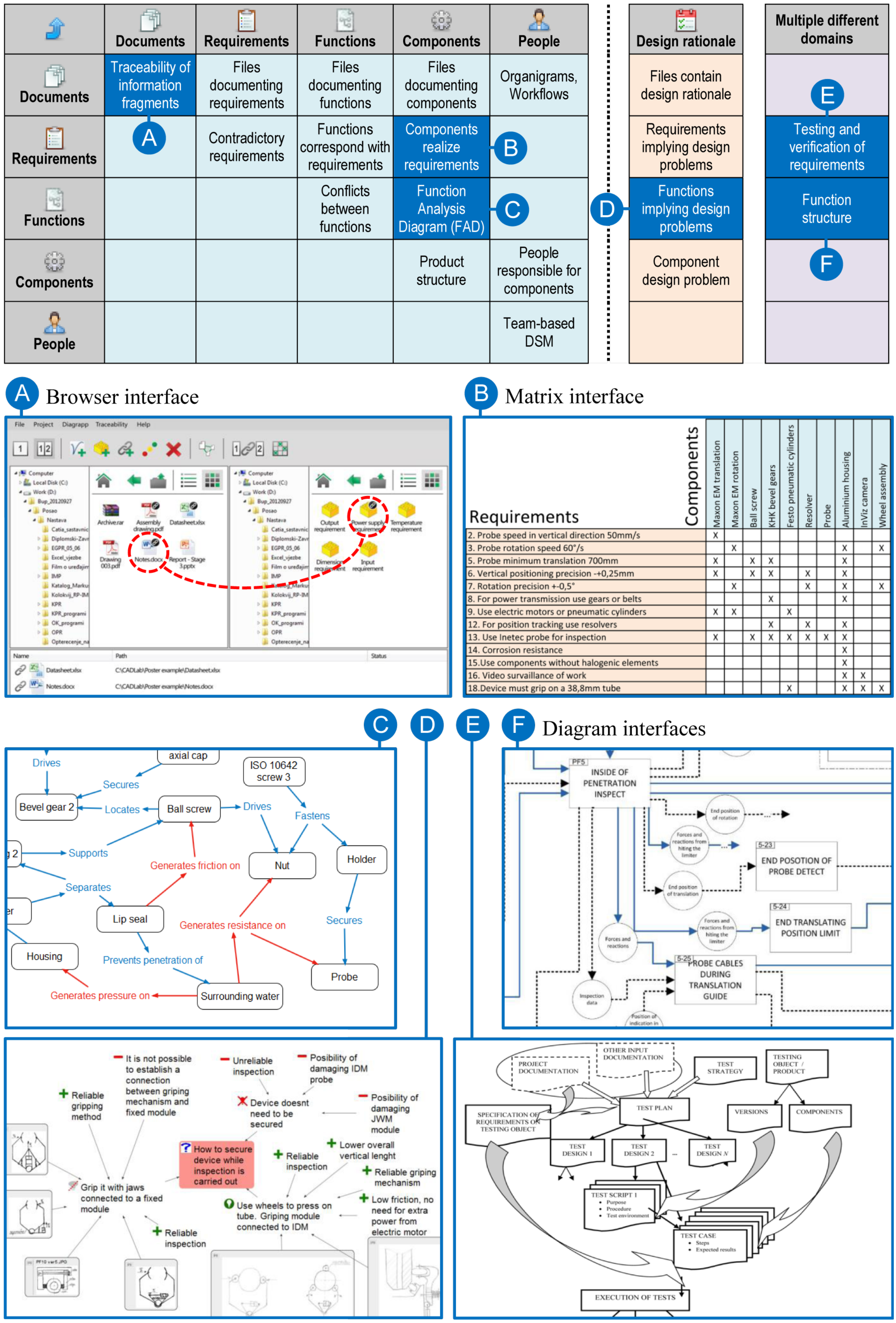

Figure 1. The basic matrix-style interface and examples of methods and tools for relation management and visualization

Some of the diagram-based methods commonly used and accepted in design practice (like e.g. IBIS diagrams and function structure) contain and visualize relations between objects from several domains. Since both of mentioned diagrams were intensively used in the monitored project, we decided to extend the basic matrix-style interface with additional columns shown in upper-right corner of Figure 1. The most dominant domain in particular "multi domain" diagram matches the row of the same domain in the matrix. We believe that such an approach could provide users a better overview in process of organizing, indexing and searching complex relation recordings and visualizations.

Presented design project has started from 24 requirements resulting with 68 functions. 276 designed components have been documented with about 1000 files including CAD files, text files, spreadsheets, catalogues, etc. Team members created 15 IBIS diagrams, 4 diagrams showing function structure and 1 function analysis diagram.

4 FINDINGS AND DISCUSSION

This section summarizes and explains the reasoning behind selection of most convenient recording and visualization interface for identified particular cases. The reasoning is mostly based on characterization of EOs, relations and domains' structure.

- For objects belonging to hierarchically structured domains the most convenient method was the browser interface (mark A on Figure 1). Examples are computer-stored files (project documentation) and product structure (components, modules, subassemblies, assemblies). Browsing is the most efficient method to access objects that exist in the same domain and are already in hierarchical relations. This hierarchical structure is too complex to be presented in form of a matrix or a diagram. Once the source and target objects are located browsing the structure, they can be selected and assigned with a relation (e.g. two project documents).
- Objects belonging to domains without hierarchical structure were interrelated using the matrix interface. Such interface also has potential of using various matrix analysis techniques (such as partitioning/sequencing, clustering, banding and tearing), sensitivity analysis techniques and coupling techniques. An inter-domain (DMM) example is the set of relations between requirements and components (mark B on Figure 1), which shows the use of specific components requested through requirement specification, or in other cases the components that satisfy requirements. Intra-domain examples are team-based and task-based DSMs, and e.g. the matrix which shows relations between functions (useful and harmful or conflicting). These matrices potentiate optimisation of people distribution, task disposition and product structure.
- For engineering objects involved in multiple complex relations (both intra- and inter-domain) we found (and we argue) that the most convenient technique is to create and visualize such a network of relations directly by drawing diagrams using the specific diagram methodology and interface tool. There are many existent diagramming methods and techniques supporting some aspects of design process whose use has been proven in engineering practice. In most cases such diagrams include engineering objects from several different domains where each object could be related in various ways to several objects from other domains and vice-versa. In presented research we tried to merge these diagrams with proposed matrix style interface concept in a way that would not change their original concepts and building methodology. One example of such diagram with relatively simple relations is Function Analysis Diagram (FAD), a method for function analysis as a form-dependent product representation (mark C on Figure 1). A FAD, unlike the Function Tree and the Function Structure, represents functions together with the physical elements of a product (Aurisicchio et al., 2013). The diagram consists of blocks used to represent product components or functions, and relations with a label used to represent either useful or harmful actions. Another example is the Function Structure (mark F on Figure 1) modelling where blocks are used to represent functions and energy/material/information flows. Third (more complex example) is the Issue Based Information System (IBIS) diagram (mark D on Figure 1). 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). The method was progressively extended by Bracewell et al. (2007) to support hyperlinking to other diagrams and computer-stored files. That way, objects of any kind could be related in IBIS diagram.

4.1 Visualization issues of established relation networks

After a set of relations between objects and in-between different domains is established, users need to be able to efficiently retrieve and expand the relationship data set. Since the relation establishment can be done through mentioned three types of interfaces, the retrieval can be done the same way – by reading (visualizing) captured data in browser, matrices and diagrams. Although browsers and matrices are efficient and straightforward when it comes to capturing and recording relations between objects, when it comes to data retrieval and traces following, diagrammatic visualization showed up to be a better solution. Visual thinking, reasoning, and analytics emphasize the role of information visualization as the powerful medium for finding causality, forming hypotheses, and assessing available evidence (Chen, 2005).

Grounded on the previous work of information traceability visualization, large networks of interrelated diagrams were created, where links between diagram files allow users to cross boundaries of a single record and browse information spread through the network, by shifting from one diagram record to another. Diagram network is automatically generated for the object selected in one of the interfaces, and represents nodes for each object that is in any way linked to the selected one, and edges for each of the relations (Martinec and Pavkovic, 2014).

The problem that needs to be tackled here is the filtering of the visualization results presented in form of a diagram. In general, displaying an entire large diagram may give an indication of the overall structure or a location within it, but makes it difficult to comprehend (Herman et al., 2000). While it is beneficial to give an insight to all the objects the selected object can be traced to, these kinds of representation can get highly complex and unclear (Figure 2).

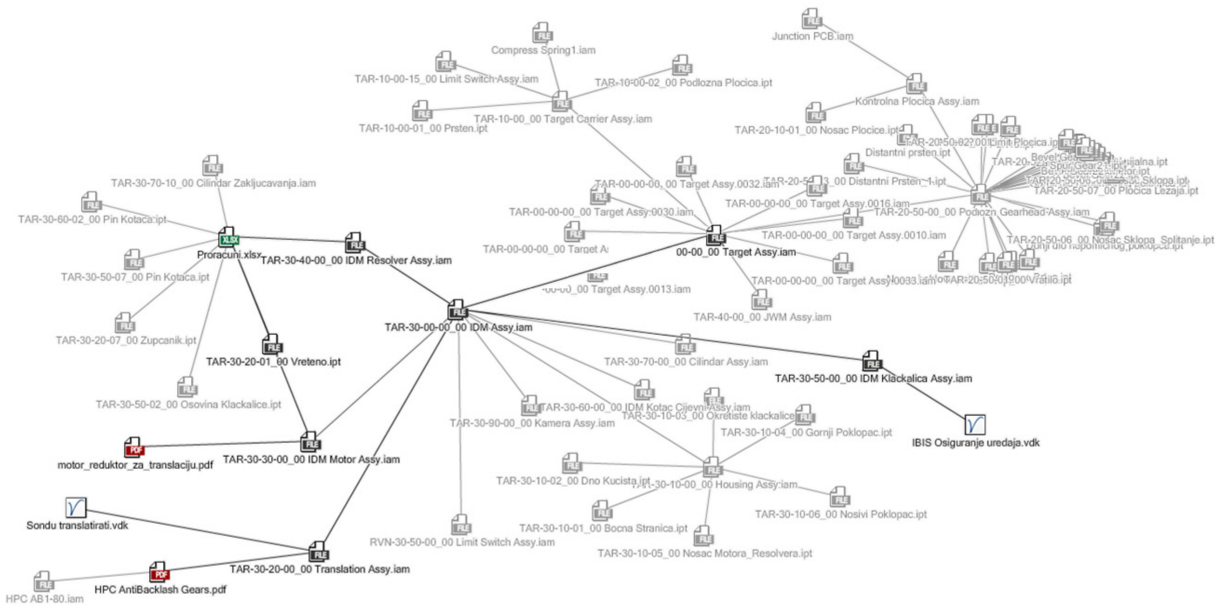

Figure 2. Only a part of the complex relation network is relevant for visualization

Herman et al. (2000) see the size of the graph (diagram) to view as a key issue in graph visualization. Besides complexity, large graphs pose several difficult problems such as performance compromise, viewing platform limits and discernment between nodes and edges, all resulting with poor usability issues (Herman et al., 2000). The issues of usability and scalability are also mentioned as two of the ten unsolved information visualization problems by Chen (2005). Therefore, comprehension and detailed analysis of data in diagrams is easiest when the size of the displayed diagram is relatively small or in this case - when the displayed data is properly filtered. Filtering the represented objects and their relations (diagram nodes and edges) should reduce the overall diagram size and complexity, thus showing only the relevant data for a particular context. Both objects and relations have to be enriched with additional attributes that are processed by the filters. Regarding objects these attributes could be for example their domain, number of relations with other objects, or some other domain-specific properties. Relations can also largely affect the resulting diagram representation. First, one has to be able to assign different types of relations between objects. A very good general proposal based on overview of several approaches and types (dependency, evolution, satisfaction, overlap, generalisation/refinement, conflicting, rationalisation, contribution, etc.) could be found in (Spanoudakis and Zisman, 2005). Relation attributes such as direction, time/date or value could also be used to filter results.

5 CONCLUSION

This paper proposes methods and interfaces for manual recording, management and visualization of relations between engineering objects from various domains. It is based on findings gathered on a monitored project and suggests interface tools and methods that are most convenient for relating objects of a particular pair or set of domains. The tools used can be divided into three basic interface concepts: browser, graph (diagram) and matrix. A framework concept has been developed where these three different methods work in synergy having an ultimate goal to create a network of relations in-between different domains. Based on experiences from our previous research and the case study presented in this paper we propose a concept of interface framework as a visualization of a matrix with multiple domains where each matrix cell leads to further procedure(s) involving either browser, matrix or diagram methods/tools.

The browser interface/tool was used to relate objects from hierarchically structured domains such as computer file system and product structure. Objects belonging to domain pairs without hierarchical structure were interrelated using the matrix interface.

For engineering objects involved in multiple complex relations (both intra- and inter-domain) we argue that the most convenient technique is to create and visualize such a network of relations directly by drawing diagrams using the specific diagram methodology and interface tool. We tried to integrate such diagrams (function structure, FAD, IBIS, etc.) with proposed matrix style interface concept in a way that would not change their original concepts and building methodology.

In order to achieve the synergy of proposed methods and tools, which will use a single, overall relation network, the existing interfaces still have to be adapted. This implies reprogramming of used browsing, matrix and diagram tools in a way that they share the same data set, so all engineering objects would be available in each interface. Thereby not only will it be easier and faster to create relation records, but also a change made in one of the tools will affect the representation of objects and relations in other tools.

Currently defined guidelines for the further research issues are:

- A filtering approach is necessary to reduce the size of the overall relation network in order to overcome the issues of complexity, usability and scalability, thus providing user only the relevant data for a particular context.
- Methodology for matrix manipulation should be improved:
  - Huge number of EO relations in large-scale projects will generate huge matrices, impossible to be manipulated and viewed as a whole – procedures and tools have to be developed that will enable hiding unnecessary areas and/or extracting and visualizing areas of current interest.
  - Semantics of relations should be added, additionally it would be beneficial if a cell would contain (or point to) more contents than just a mark of relation existence.
  - Mechanisms (procedures) for generating and inserting predefined templates of selected matrix areas should be developed and implemented.
- Monitored design project finished with detail design phase, but we will ask design team to analyse the utility of recorded relations and their visualization in future similar design projects. Also it remains to capture and visualize objects and relations in process of prototype testing and requirement verification (mark E on Figure 1).
- Proposed framework has not been considered in context of PLM environment - embedding and merging this proposal with PLM system has to be one of the major lines for further upgrades.
- Presented case study has been conducted in medium sized company. It would be necessary to repeat the study in large company, e.g. in automotive sector, to explore and compare the differences in domains and collections of engineering objects as well as their characteristics.
- It should be verified if (and how) proposed concept of relations between domains (cells in matrix on upper part of Figure 1) contributes to building of shared understanding of design projects on the company level. These cells could be considered as predefined contexts for relation recording as well as guidance for information and knowledge retrieval. We consider mentioned "matrix" (its concept and proposal of particular contexts) as the major contribution of this paper.

REFERENCES

Aurisicchio, M., Bracewell, R. and Armstrong, G. (2013) The function analysis diagram: Intended benefits and coexistence with other functional models. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 27, Special Issue 03, pp. 249-257.

Aurisicchio, M, Bracewell, R. (2013) Capturing an integrated design information space with a diagram based approach, Journal of Engineering Design, Vol. 24, No. 6., pp. 397-428.

Beier, G., Figge, A., Müller, R., Rothenburg, U. and Stark, R. (2013) Supporting Product Development through Cross-Discipline Dependency-Modeling – Novel Approaches for Traceability-Usage. Lecture Notes on Information Theory, Vol. 1, No. 1, pp. 21-28.

Bracewell, R., Gourtovaia, M., Wallace, K. and Clarkson, J. (2007) Extending design rationale to capture an integrated design information space. Int. Conf. on Eng. Design, ICED ’07, Paris, Design Society.

Chen, C. (2005) Top 10 unsolved information visualization problems. Computer Graphics and Applications, IEEE, Vol. 25, No. 4, pp. 12-16.

Herman, I., Melancon, G. and Marshall, M.S., (2000) Graph visualization and navigation in information visualization: A survey. Visualization and Computer Graphics, IEEE, Vol. 6, No. 1, pp. 24-43.

Hurwitz, J. and Kaufman, M. (2007) Leveraging information for innovation and competitive advantage. Hurwitz & Associates White Paper, www.hurwitz.com.

Keller, T. and Tergan, S.O. (2005) Visualizing Knowledge and Information: An Introduction. Lecture Notes in Computer Science: Knowledge and Information Visualization, Springer.

Koehler, N., Naumann, T. and Vajna, S. (2014) Supporting the Modeling of Traceability Information. Proc. of the 13th Int. Design Conf., DESIGN 2014, Dubrovnik, Croatia, pp. 1811-1820.

Koenigs, S.F. (2012) Konzeption und Realisierung einer Methode zur templategestützten Systementwicklung. Doctoral Dissertation, Technical University of Berlin, Germany.

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

Lindemann, U., Maurer, M. and Braun, T. (2009) Structural Complexity Management. Springer.

Marcus, A., Xie, X. and Poshyvanyk, D. (2005) When and how to visualize traceability links?. Proc. of Third Int. Workshop on Traceability in Emerging Forms of Software Engineering, Long Beach, USA, pp. 56-61.

Marjanovic, D., Storga, M., Bojčetic, N., Pavkovic, N. and Stankovic, T. (2011) EUREKA E4911 TRENIN Project Final Report. www.trenin.org.

Martinec, T. and Pavkovic, N. (2014) Visualization of Information Traceability In Product Development. Proc. of the 13th Int. Design Conf., DESIGN 2014, Dubrovnik, Croatia, pp. 1831-1842.

Maurer, M.S. (2007) Structural Awareness in Complex Product Design. Doctoral Dissertation, Product Development, Technical University of Munich, Munich, Germany.

Pavkovic, N., Bojcetic, N., Franic, L. and Marjanovic, D. (2011) Case studies to explore indexing issues in product design traceability framework. Proc. of the ICED 11 Int. Conf. on Engineering Design 6, Copenhagen, pp. 131-140.

Pavkovic, N., Tečec Ribarić, Z. and Sviličić, T. (2012) Traceability Case Study on Rail Vehicle Control Unit Development Project. Proc. of the 12th Int. Design Conf., DESIGN 2012, Dubrovnik, pp. 1567-1578.

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

Salustri, F.A., Eng, N.L. and Weerasinghe, J.S. (2008) Visualizing Information in the Early Stages of Engineering Design. Computer-Aided Design & Applications, Vol. 5, No. 5, pp. 697-714.

Spanoudakis, G. and Zisman, A., (2005) Software traceability: a roadmap. Handbook of Software Engineering and Knowledge Engineering, Vol. 3, pp. 395-428.

Wickel, M.C., Schenkl, S.A., Schmidt, D.M., Hense, J., Mandl, H. and Maurer, M. (2013) Knowledge structure maps based on Multiple Domain Matrices. InImpact: The Journal of Innovation Impact, Vol. 5, No. 1, pp. 5-16.

ACKNOWLEDGMENTS

This paper reports work funded by Ministry of Science, Education and Sports of the Republic of Croatia, and Croatian Science Fundation MInMED project (www.minmed.org). Presented case study analysis is conducted in cooperation with INETEC d.o.o., a company that develops technologies for nuclear power plant examination and repair.