INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

TRACEABILITY CASE STUDY ON RAIL VEHICLE CONTROL UNIT DEVELOPMENT PROJECT

N. Pavković, Z. Tečec Ribarić and T. Sviličić

Keywords: traceability, HW and SW testing, engineering documentation management

1. Introduction

In order to master challenges of the modern manufacturing paradigm and confront with the challenges of the complex products and services, companies have recently provided new approaches for reusability, adaptability, and variety of products, services and engineering information.
The importance of engineering information is underlined by the fact that product lifecycle viewed as chain of information transformation processes both consume and create large amounts of information as they proceed. During the different stages of the product lifecycle different participants will acquire information from many sources, such as handbooks and design guides, catalogues, journals, books, training courses, previous projects, discussion with colleagues and customers, user and service guides, disposal reports, etc. In order to fully understand an item of information it is necessary to know something about the circumstances in which it has been developed and recorded.
Traceability of information provides the basis for assessing the credibility of engineering information, its better understanding and making judgments about the appropriateness of its use for a particular task [Štorga et al. 2009a]. Traceability has been considered as a quality attribute and many standards governing systems development require the creation of traceability procedures. Currently there is little provision for acquiring, capturing and delivering with the engineering information, the information that provides its development context, and few tools to support this process. In addition, little is currently understood about the requirements for engineering information traceability in product design and development environment, and there are few methods by which effective traceability can be ensured [Štorga et al. 2009b]. The work reported here builds on the TRaceability of ENgineering INformation - TRENIN (www.trenin.org) framework for engineering information development traceability [Marjanović et al. 2011] by discussing the outcomes of case study conducted in one of TRENIN project industrial partners.
Traceability is the ability to verify the past, location, or application of an item by means of documented recorded identification. The existing practice of recording the outcome of the engineering design process is almost exclusively based upon highly formalised model of the product, in the form of computer-aided engineering models, bills of materials, engineering change orders, etc. However, the detailed process, activities and rationale by which the design has been created and the engineering design information (EDI) developed (to the extent that they are recorded at all) are recorded largely in an informal manner [Giess et al. 2007]. A consequence is that is difficult to retrace or audit the engineering reasoning that has taken place during the process of EDI development without extensive work to assimilate and digest design documentation, and that identification of relevant parts of the information records within the documentation requires significant skill and often an intimate knowledge.

Traceability may also be defined as the ability to help stakeholders understand the associations and dependencies that exist among entities created or used during a product development process.
This paper presents results of first implementation of traceability methodology and prototype tool in design office of TRENIN project industrial partner. Proposed traceability framework is thoroughly described in [Marjanović et al. 2011], [Štorga et al. 2011a] and [Štorga et al. 2011b].
The purpose of this case study was to give us the first feedback on usability and efficiency of proposed traceability framework and methodology by answering following (research) questions:

1. What are the situations that trigger the requirements for tracing?
2. Which are most common "starting points" for tracing in current engineering practice – how they could be structured?
3. What is most often being looked for, and what is expected to be found?
4. How to structure information fragments in information objects regarding to various contexts and phases of product development process?
5. How to represent and record informal information in traceability process?

The remainder of this paper is structured as follows: a brief overview of proposed traceability framework as the basis for understanding of case study methodology and results. A description of HW and SW testing process specific for industrial partner which participated in case study is given to illustrate the proposed traceability records structure. Last chapter of paper discusses possible utilization scenarios of proposed traceability records as well as conclusions drawn from case study, together with directions for further research and TRENIN methodology improvements.

1.1 Related work

Because of their impact and importance, requirements management and traceability has been in the focus of many researches both in mechanical design and in software design. Most designs are being evaluated according to fulfilment of requirements. Therefore backtracking from final design characteristics, solutions and components to requirements and vice versa – tracing from requirements to final design is essential. By capturing the information how components realize various functions and which requirements are mapped to different components, the engineers would be able to verify results of the product development. Gotel and Finkelstein [Gotel and Finkelstein 1993] investigated and discussed the underlying nature of the requirements traceability problem. Gausemeier et al. [Gausemeier et al. 2010] showed how to manage complex dependencies within the principle solution of a mechatronic system by using the example of requirements tracing. The authors considered relations in the principle solution, which are substantial for the trace of requirements.
Traceability in software engineering has got more attention of researchers than in engineering design. Several models and methodologies were developed, mainly focused on requirements traceability and related issues – [Mohan et al. 2008], [Ramesh and Jarke 2001]. An example of comprehensive research projects in this area is the "MOST" project (http://most-project.eu). Schwarz et al. [Schwarz et al. 2008, 2010] present the approach that supports the definition of metamodels for traceability information, recording of traceability information in graph-based repositories, identification and maintenance of traceability relationships using transformations, as well as retrieval and utilization of traceability information using a graph query language. Burge & Brown [Burge and Brown 2007] describe how their system supports requirements traceability in software engineering by incorporating functional and non-functional requirements into the argumentation for design rationale.

2. Overview of proposed traceability framework

Research work in TRENIN project led us to development of separate abstract level for building the representation of information object (IO) relationships and putting them together around (in) specific context. In such representation each IO could be repeatedly represented in various contexts. Various contexts may contain different subsets of information objects and their relationships. A concept of representation of such "context driven" subset of product documentation is named "Traceability Record" [Štorga et al. 2011b]. Besides "context modelling", the main purpose of "Traceability Record" (TR) is to be a container for recording traces of information objects development through specified time period – named "a design episode". Representation (definition) of specific context is

realised by extracting the subset of "product development ontology" developed as essential part of the whole traceability framework. In other words, Traceability Record identifies physical and abstract concepts and relations from the product development domain relevant for description of the information object in the context of the product development process. Figure 1 shows the basic idea (concept) of traceability records.
Subset of ontology is indicated as "TE" nodes (Traceability Elements). Traceability elements are representations of notions and entities from design process (information) and product (physical) domains. When particular design episode is finished, associated traceability records are being stored in database, enabling rich mechanisms for back tracking, interpretation, searching for information and information origins, etc. in the environment similar to semantic network.

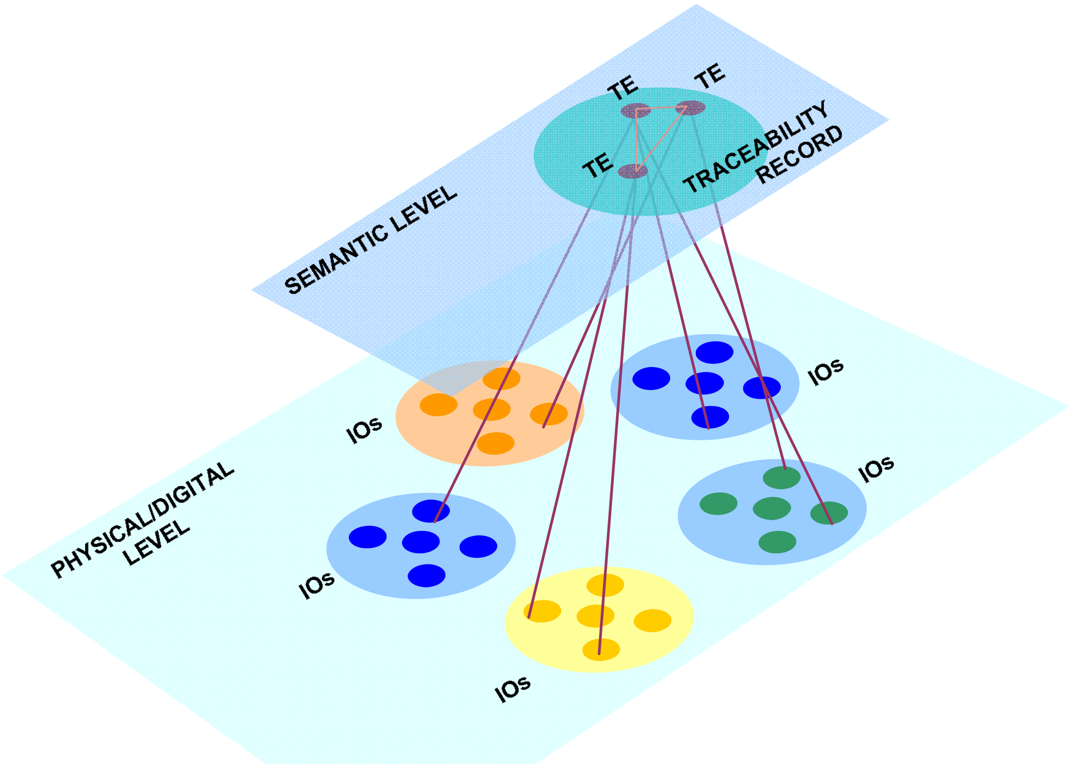

Figure 1. Schematic overview of "traceability record" concept

2.1 Product development ontology

We are developing and testing a concept where a subset of ontology will define a broader and richer context for indexing and tracing the complex set of information objects, designers, their actions and situations in design process history. In this approach, ontology denotes a conceptual data schema that represents the relevant domain entities and their relationships by means of classes and relations. Elements of ontology subset are associated with information objects (in most cases design documentation) belonging to design episode which is to be traced (Figures 2 and 3). TR is a continuously updatable and accessible data structure that evolves in parallel with referenced information objects being traced during specific product development episode. Traceability episode covers a certain time interval and/or sub process in product development process. Episode is composed of sequence of traceability events which represent key events and sets of activities and/or milestones in product development process, usually prescribed within organizational and process workflow and modelled in existing software tools. Tracing procedure is focused (but not exclusively) on events that are the part of the process of information object management in PLM system and user generated events that are not manageable by PLM. Part of system architecture named "traceability engine" is responsible to recognize that associated event happened in particular point of the traceability episode. In other words, each traceability record should be associated to the set of

predefined events that trigger the process of capturing and recording traceability data and information object development.
Prior research [Štorga et al. 2009b] suggests that a traceability framework will comprise a top level ontology for definition of traceability record templates. This set of traceability record templates can be customized and extended within the scope defined by the ontology accordingly to the specific traceability needs. The idea is that traceability manager in particular organization selects relevant parts of the traceability ontology, expands them accordingly to the traceability needs in particular organization, and create set of traceability record templates that could be used in specific traceability episodes. In presented research, we decided to adopt a Merged Ontology for Engineering Design (MOED) [Ahmed and Štorga 2009], as a top level ontology for definition of the TRENIN formal language.
In proposed TRENIN ontology two main areas have been distinguished: traceability elements (TE) and traceability objects (TO). Traceability elements represent (mainly abstract) concepts extracted from design process domain while traceability objects represent physical and digital entities from design organization and management domain, with special focus on design documentation. Relations in proposed ontology are named traceability links (TL).
Traceability Elements (TE), are on traceability records associated to the information objects managed by external applications represented with traceability Objects (TO) e.g. projects, products, documents, items, users, flow processes, etc. Relations between traceability objects should represent dependencies between content, hierarchy, timelines, etc. To achieve full traceability, it is necessary to extend the model with fragmentation of IO content. Traceability Events (TEV) are driving execution mechanisms of the tracing – recording procedure. TEV represent events on information objects managed by external application (e.g. new, release, approve, update, etc.) or are generated manually by the user during the traceability episode. Figure 2 is an example of one particular traceability record scheme – each of four interrelated traceability elements has several associated traceability objects – all together these elements compose the particular context for tracing.

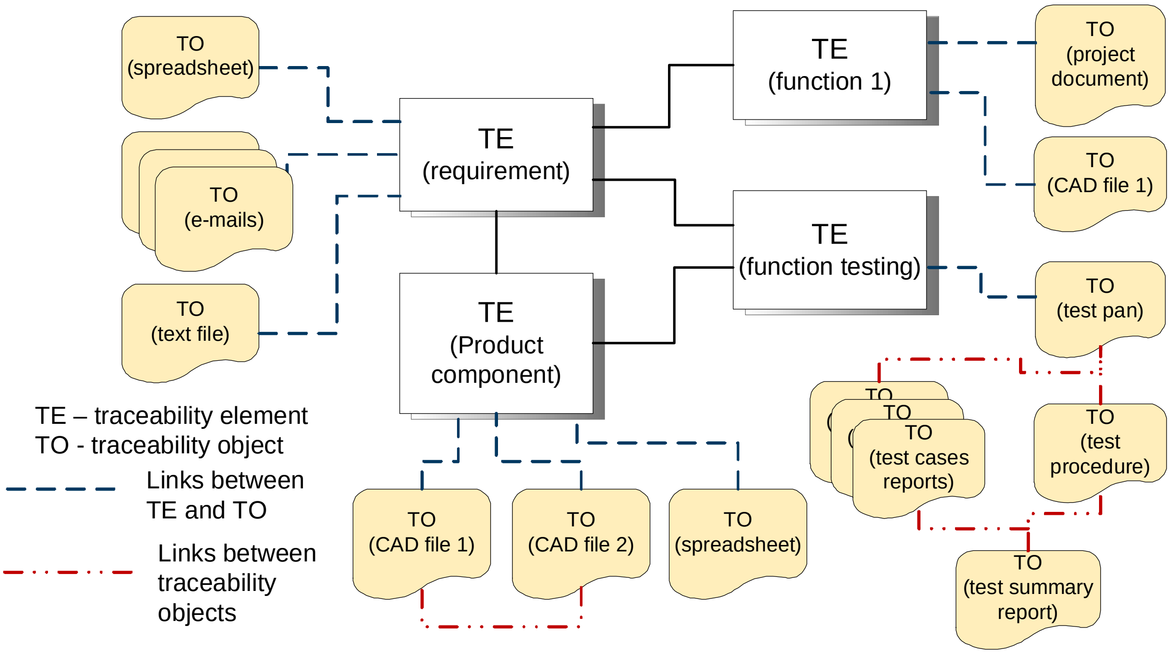

Figure 2. An example of traceability record scheme

2.2 Traceability methodology and usage scenario

Considering how to achieve traceability in product development, we can emphasize the existence of the three main stages whose description follows.

1. Identification and planning stage is characterized by definition of traceable items accordingly to design episode where the designer decided to make things traceable. In this stage, the main task is to define which objects should be traced and what kind of links are needed between those objects.
2. Recording and documentation - the main task of this stage is in creation of traces that are result of product development activities, designers' actions, decisions, reasoning, events, etc. Those design items (information objects origins and evolution) and design routes that have been explicitly defined in traceability record should be recorded and documented for further use. Through the timeline of development project realisation, the traceability record is being updated, upgraded and "filled" with traceability data. The result of the development project episode is finalised traceability record stored in database.
3. Utilization phase has to provide understanding and reuse information in right context by:

- answering the complex questions that arise in later phases of product life cycle
- reconsidering the product development history either for this product development issues or for other similar products
- finding the source (origin) of mistakes or troubles
- simulating design episode in another situation - for performing changes on existing solutions, reusing of the existing solutions in new projects, configuration of the new variant of the product, educational process for inexperienced designers

3. Traceability case study

This chapter presents the results of case study of TRENIN methodology and prototype tool conducted in TRENIN industrial partner – "Končar electrical engineering insitute Zagreb" (KIEE) – the department for power electronics and control. This department develops and produces control systems for general industrial purposes and especially for power plants and railway vehicles. The case study started with thorough study of the technical documentation structure for several finished major products. There were no possibilities to conduct case study on ongoing project, therefore we have decided to conduct a case study on finished development project. A part of the development process of complex mechatronic product has been chosen - 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.
This project is part of development of new train for regional traffic. The development process of VCU comprises a lot of HW and SW design and testing, as well as mechanical design of housing. Identified product development process structure, documentation and product hardware and software components have been mapped to proposed ontology. In other words, each entity identified in mentioned project has been mapped to new instance of particular notion in ontology. This way, an extension of ontology has been developed - valid for KIEE department for power electronics and control. New instances and ontology concepts were stored in MS Access database tool that was used to explore TRENIN utilization issues. The traceability paths were identified and simulated by the researcher following the particular development process [Sviličić 2011].
In this case study we have focused on two different phases and contexts of the project:

1. Early development phase - consolidation of project and product requirements – main deliverable of this phase is the detailed product development plan. Here we have been focused to requirements traceability.
2. The process of HW and SW testing in late development phase.

3.1 Tracing of early development phases

To trace the engineering information development process in early phases of chosen project we have developed a traceability record named "Consolidation of project and product requirements". Detailed structured list of this traceability record is shown on Figure 3. The purpose of this record is to trace the creation of main project documents – project development plan, project Gantt chart and VCU technical data. These documents contain all the information about project definition – list of requirements, goals, resources, references, task distribution list, product component structure, etc.

Mentioned three documents are instantiated as traceability objects – on Figure 3 they have ID's 75, 87 and 5 respectively. These three documents are expected to be created after the traceability record will be activated – therefore their "new" check box is turned "on". Main project document "Product development plan" is associated to element "product planning", Gantt chart is associated to element "management activity – scheduling", and the document with VCU technical data is associated to newly created instance of element "defining" – "defining VCU technical data". Traceability element "requirement gathering" is used to index several documents that are sources of requirements and other data necessary in creation of before mentioned three main project documents. These documents are relevant standards and contract(s) (and offers) with other partners in development process of new regional train. To complete the developed traceability record context, meeting minutes from VCU development team initial meetings are also included, associated to element "decision making".
Figure 3 is also an example of basic database "report view" of one traceability record which is a list of traceability elements and their associated information objects. Each element and object has a button which activates further tracing possibilities and shows all basic data (author(s), creation date etc.), for that particular object or element.

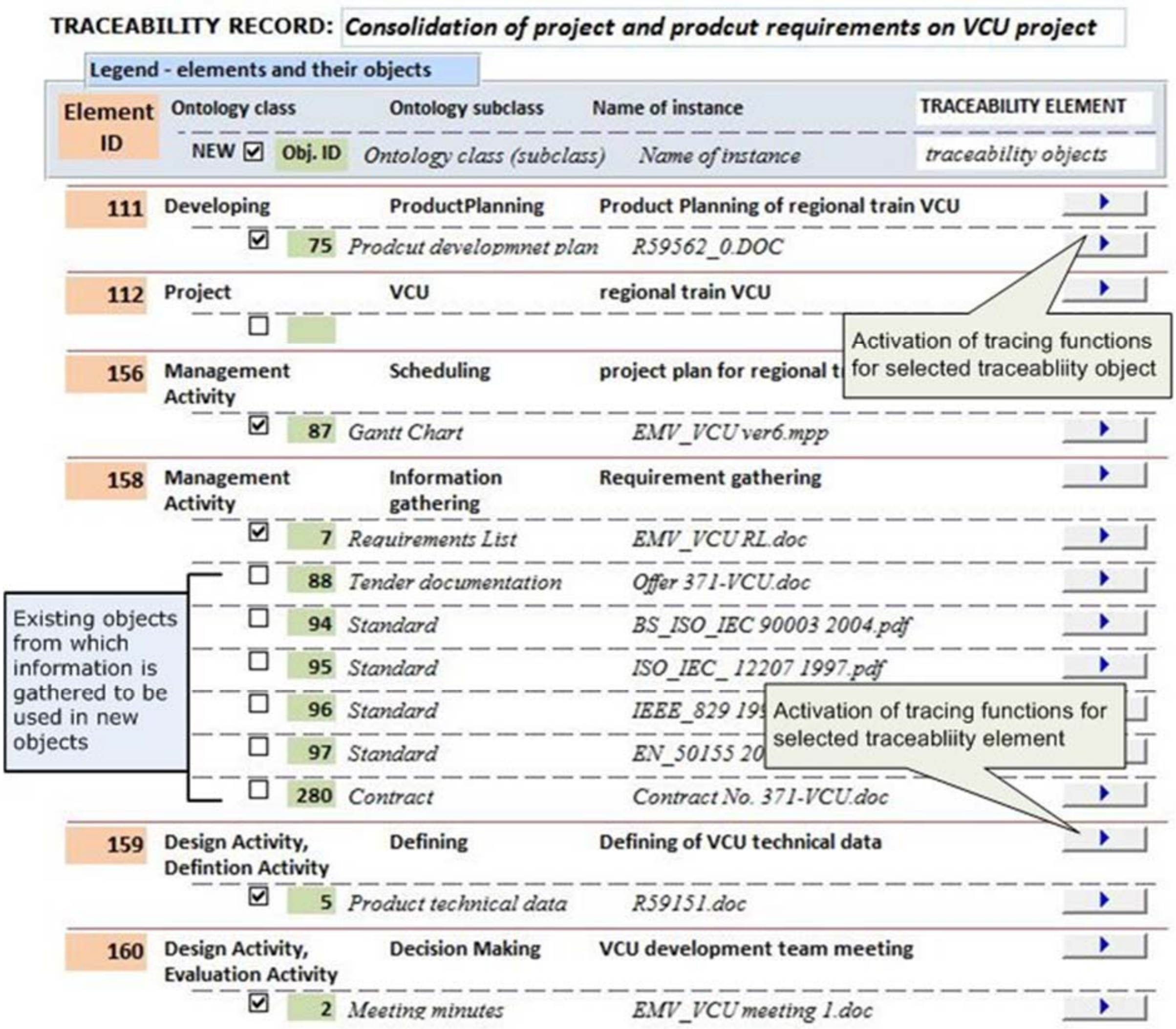

Figure 3. Traceability record for tracing of project consolidation phase

For selected traceability element the user could generate the list of all relationships with other elements on this traceability record, as well as relationships of this element with other elements on other traceability records. To start a broader search, user could also get a list of all traceability records containing selected element. Similarly, for selected traceability object system offers a list of PLM

event history, relationships with other objects, other records containing this object, all other objects of the same class, etc.
By combining mentioned actions user is able to navigate through search space of stored traceability records and their content, trying to find desired data or answers to complex questions.

3.2 Tracing of HW and SW testing process

This product development phase (specific for control systems) has been chosen as an example of process that generate a huge set of relatively small documents in which many authors from several different departments (employees) have participated. Documents are highly interrelated from various viewpoints – hierarchically, by evolution of content and by referring to content fragments. The testing process is also very interesting from the viewpoint of utilization, especially from the viewpoint of knowledge reuse. Figure 4 displays a defined documentation structure which has to be created in testing process, according to IEEE 892:1998. Documents of testing strategy must be defined for each type of product/product component. It contains general guide lines for test performing as well as test techniques for each type of product/product components. Test plan is the basic document which has to be defined before testing process, on the basis of testing strategy, input project documentation and requirements specification.
Test Case is actually a test of one or more required device functions – test case is the basic item of testing process. In each test following features has to be defined:

- product functions/requirements that are being examined,
- product components (both HW and SW) which realize these functions/requirements.

Table 1. List of all document classes that are being generated in testing procedures

Test strategy	Defines testing structure, testing rules and instructions for test plan writing. One test strategy is valid for one type of product/product component.
Test plan	List of requirements associated to product/product components. Has to be generated before testing for each project/product.
Requirements specification	Usually this is a part of project (manufacturing) documentation.
Test design specification	Document which specifies which product/product component will be tested, which approaches will be used and which test cases will be included
Test script specification	This document defines the purpose of testing and the detailed description of testing environment.
Test case procedure	For each test case, test steps and expected results have to be defined.
Test log	Results of test case performing. Test resulting values have to be predefined in test case procedure (Pass, Fail, Acceptable, Not tested). Comments and diagrams have to be attached.
Test incident report	Has to be generated if failures or incidents happen during testing.
Test summary report	Main document about product testing results.

Figure 4 and Table 1 give an overview of a structure of documentation being generated during the HW and SW testing. This is a huge set of documents with complex hierarchical structure, also with many other dependencies and data flows between documents. Many different authors participate in this process having different roles in particular phases. Having all these facts in mind, we considered this process as very suitable for developing several more complex traceability records. Testing results, especially test incident reports are very interesting candidates to be back-tracked in utilization process.

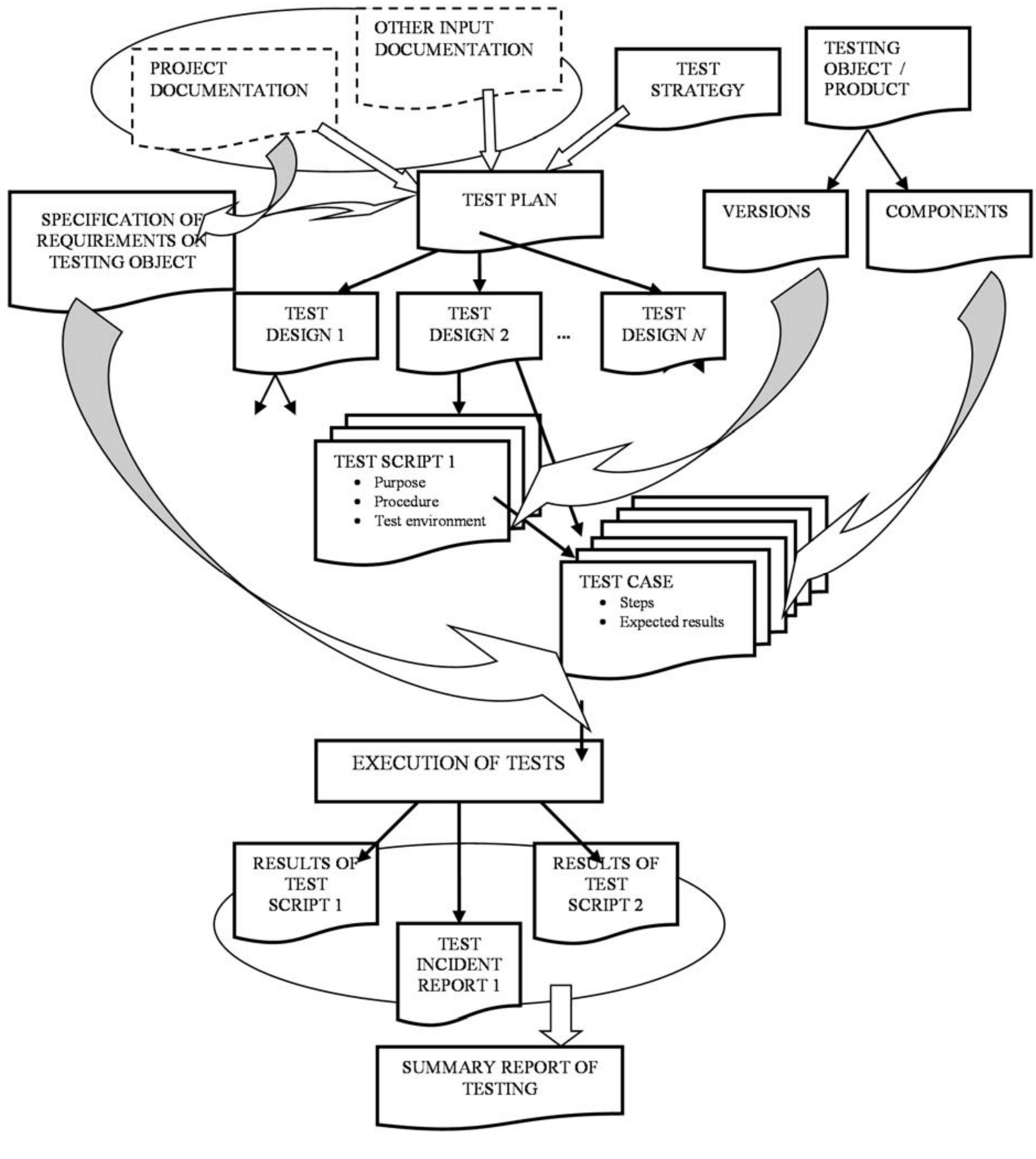

Figure 4. Testing documentation structure

For each conducted test it is necessary to write a report and a test incident report if incident happen. Several elementary tests form a test script. Test script defines a test environment. In case of changing the version of some component, some or all of tests have to be done again. Several test scripts form a test design. An overview of all testing document classes is given in Table 1.

3.2.1 An example of utilization situation

Let us describe one of the most probable utilization situations which can occur in future development projects.

A component from previous project is being reused in current (future) project, but the testing process reports some kind of error or unexpected problem. TRENIN user(s) will try to find all the testing documentation of previous testing procedure for that component in similar environment. After identification of all products where this component was used, the next step would be to find testing reports for the same function that fails in current testing. Now the user has extracted a set of links to documents where he/she will try to find if a similar problem has occurred in previous component usage and find an explanation of causes of problem and solutions. If this component worked perfectly in previous testing, at least the TRENIN user could collect all the data about environment conditions of previous component usage (interaction with other components, SW version, operating system version, authors of SW modules, etc).

It is very difficult to predict what the user will need in such examples of future situations. Therefore we have developed four variations (form different viewpoints) of traceability records for tracing VCU testing procedure. Only further research and intensive usage of prototype tool (at least for two years) could give us the answer which concept of building traceability record could give the users the best results in utilization.

For complex utilization questions most probably user(s) will have to combine traces and/or information extracted from more than one record to reach the desired answers. This concept of "walking through" (or browsing) is the main mechanism and idea of utilization tool usage.

Here we will briefly describe the concept and content of each of these four records. Complete list of structure and content of these records are too long to be listed in the paper.

3.2.2 Traceability record "Component design"

The primary purpose of this record is to indicate which component realises which product function. Each function and each component are instances of traceability elements. Relations between these elements indicate which component realises which function. In case of VCU device there are about 30 functions, and in most cases one component realises one function. Therefore this traceability record contains relatively big set of elements, and just few traceability objects. This record's purpose is rather to provide the initial information and/or starting point in utilization (especially for novice designers) than to trace the information development. Further research could be directed towards development of general templates of such concept of record for various products or classes of products.

3.2.3 Traceability record "Testing - generally"

The purpose of this record is to associate manufacturing documentation with each component as the references (inputs) for defining test strategies and test plans for each particular component. This record could be useful in developing test plans and strategies, as well as to trace changes of versions or updates in manufacturing documentation. Version changes and updates occur frequently, mostly because some of electronic components become obsolete during development period and are replaced by similar ones. Updates are even more frequent in SW components. This concept of record also enables the user(s) to get a clear picture of environment (HW and SW versions) in which the particular testing took place. As discussed before, such information is essential for utilization situations where user wants to find the cause of current problem by searching previous similar problems for reused components.

3.2.4 Traceability record "Testing of DMK 126 module - complete testing process"

This record focuses to complete testing process (to lowest level documents) of one single component. Component "DMK 126" is chosen as suitable example because this is one of the components that realise more that one function. Here the traceability objects are all kinds of the documents that are described in Table 1. Further research questions that arise here is how to extend the current TRENIN model to eventually group several traceability objects (like e.g. 15 test logs associated to one element) to make the record creation process more efficient.

3.2.5 Traceability record "Testing of DMK 126 module – system failures"

This record focuses only to system failures happened in one single component testing process. This approach tends to integrate previous three contexts around test incident reports for one single component. Such approach is focused on only one utilization issue which enables TRENIN users to reduce search in seeking for specific answers. On the other way, this approach could be paid off only if such specific utilization issues are frequent enough in practice.

4. Discussion and conclusions

Predefined and organized reuse of stored traceability data in a proposed TRENIN system could become an integral part of the design project solving process. Predefined and organized reuse could be achieved through development of templates of traceability records focused to more frequent utilization issues and situations. Some simple examples of these situations could be: changes on existing solutions, reusing of the existing solutions in new projects, viewing problems and solutions on previous similar projects, analysing testing reports, etc. Eventual implementation of TRENIN system in particular environment should start with analysis of potential utilization issues and extracting most frequent of them. Firstly created traceability records should be focused to these most frequent utilization issues – that way the benefits for users could be achieved as fast as possible - in order to motivate the employees to use the system.

The conducted case study showed that proposed TRENIN model gives endless possibilities of combining elements, objects, tracing time periods and contexts (wider or more focused) in creation of traceability records. To achieve reasonable benefits from the system, a careful planning of step by step process is necessary in the first phase of system implementation.

Previous experiment with taxonomy-based indexing [Pavković et al. 2011], as well as this case study gave us an insight in problems that may occur in process of initial definition and selection of ontology instances of traceability elements. It is very difficult to propose one common ontology structure that will equally suit the needs of all participants in particular product development process [Ahmed and Štorga 2009]. 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 ontology structure. Due to ambiguity of abstract categories developed through examination of engineering knowledge using top-down approach, we argue for "bottom-up" approach where hierarchy is being built from leaf and/or lower level taxonomy nodes which in fact represent the end solutions - actual or concrete terms that could be easily defined and understood by designers.

Based on proposed approach to defining context for tracing and indexing with traceability records that are subsets of ontology, the knowledge integration in utilization process could be accomplished in two ways:

- using the existing relations in ontology to navigate (perform semantic search) between related elements of several traceability records (which define different contexts),
- using links to traceability objects and elements to "walk through" various records – building new traces or following existing traces that were recorded during record active tracing episodes

Both ways should lead towards building a complex navigable semantic network structure.

Conducted case study also gave us directions for further improvements of TRENIN model and methodology:

- the most important improvement could be to develop a model of fragmentation of the information and information flow among information objects;
- to include and model the workflow as an traceability object and the driver of the traceability execution:
- relationships between information objects should be included in model, as well as reference lists for object contents (and data flow) should be modelled;
- in future development more attention should be paid to traceability of elements (ontology concepts) – focusing only on information objects traceability is not sufficient to achieve full traceability;
- traceability of the content and context across different traceability records – how to connect records with predefined mechanisms that will ease utilization process;

- user interface(s) for record creation, execution and utilisation should be further developed to achieve desired level of efficiency.

Acknowledgments

This paper reports work funded by EUREKA E!4911 TRENIN project (www.trenin.org) and Ministry of Science and Education of Republic of Croatia.

References

Ahmed, S., Štorga, M., (2009). Merged Ontology for Engineering Design: Contrasting an empirical and a theoretical approach to develop engineering ontology, AI EDAM: Artificial Intelligence for Engineering, Design, Analysis and Manufacturing. Vol.23 No.4, pp. 391-407.

Burge, J., Brown, D. C., (2007). Supporting Requirements Traceability with Rationale, GTC'07: International Symposium on Grand Challenges in Traceability, Slade, KY.

Flanagan, T. L., Eckert, C. M., Clarkson, P. J., (2003). Parameter trails. Proceedings of the 14th International Conference on Engineering Design (ICED'03), Stockholm.

Gausemeier, J., Dorociak, R., Pook, S., Nyßen, A., Terfloth, A., (2010). Computer-aided cross-domain modeling of mechatronic systems. Proceedings of international design conference - DESIGN 2010, pp. 723-732.

Giess M. D., Goh, Y. M., Ding, L., McMahon, C. A., (2007) Improved product, process and rationale representation and information organisation to support design learning, International Conference on engineering design ICED '07, Paris

Gotel, O. C. Z., Finkelstein, C. W., (1993). An Analysis of the Requirements Traceability Problem. Proceedings of the First International Conference on Requirements Engineering, Colorado Springs, CO, USA, pp 94 – 101.

Kleinsmann, M., Buijs, J., Valkenburg, R., (2010). Understanding the complexity of knowledge integration in collaborative new product development teams: A case study. Journal of Engineering and Technology Management, 27, pp. 20–32.

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

Mohan, K., Ramesh, B., (2007). Traceability-based knowledge integration in group decision and negotiation activities. Decision Support System, Elsevier, Volume 43, pp 968-989

Mohan, K., Xu, P., Cao, L., Ramesh, B. (2008). Improving change management in software development: Integrating traceability and software configuration management. Decision Support Systems, 45, 922–936.

Pavković, N., Bojčetić, N., Franic, L., Marjanović, D. (2011). Case studies to explore indexing issues in product design traceability framework. Proceedings of the ICED 11 International Conference on Engineering Design, Copenhagen, Vol. 6, pp. 131-140.

Ramesh, B., Jarke, M., (2001). Toward reference models for requirements traceability. IEEE Transastions on Software Engineering. 27(1), 58–93.

Schwarz, H., Ebert, J., Riediger, V., Winter, A. (2008). Towards querying of traceability information in the context of software evolution. 10th Workshop Software Reengineering (WSR 2008).

Schwarz, H., Ebert, J., Winter, A. (2010). Graph-based traceability: a comprehensive approach. Software Systems Modelling, Vol. 9, No 4, 473-492.

Štorga, M., Darlington, M., Culley, S. J., Marjanović, D., (2009a). Traceability of the development of 'information objects' in the engineering design process. Proceedings of the 2nd International Conference on Research into Design ICoRD'09, Bangalore, India.

Štorga, M., Darlington, M., Culley, S. J., Marjanović, D., (2009b). Toward a process and method for tracing the development of information objects used in engineering design. Proceedings of the ICED 09 Int. Conf. on Engineering Design, Stanford, USA.

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

Štorga, M., Pavković, N., Bojčetić, N., Stanković, T., (2011a). Traceability of engineering information development in PLM framework. Proceedings of 8th International Conference on Product Lifecycle Management – PLM'11, Technical University of Eindhoven, The Netherlands.

Sviličić, T., (2011), Traceability of engineering information and documentation, Master thesis, Faculty of mechanical engineering & naval architecture, University of Zagreb, (in Croatian)

Dr. Neven Pavković
Associate Professor
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture
Ivana Lučića 5, 10000 Zagreb, CROATIA
Telephone: +385 1 6168 545
Telefax: +385 1 6168 284
Email: neven.pavkovic@fsb.hr
URL: http://www.cadlab.fsb.hr