Facilitating design communication through engineering information traceability

Q1 NEVEN PAVKOVIĆ, MARIO ŠTORGA, NENAD BOJČETIĆ, AND DORIAN MARJANOVIĆ
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Zagreb, Croatia

Abstract

Traceability of information provides the basis for assessing the credibility of engineering information, better understanding it, and making judgments about the appropriateness of its use for a particular design task. The presented research attempts to answer how the proposed traceability methodology and framework could help designers to improve communication, eventually create new channels of communication, and contribute to the creation of shared understanding in collaborative design processes. The discussion of these issues is based on a literature review, empirical research, observations of industrial practice, and feedback from initial implementation. The research is focused on information objects (IOs), archetypically represented in the engineering domain as technical documents that are often complex structures constituted of textual, numerical, and graphical fragments. The presented approach is based on an abstraction of IOs' relationships organized around specific contexts that are defined by a subset of product development ontology. Each IO could be repeatedly represented in various contexts that may contain different subsets of objects and their relationships. Such a representation also acts as a container in which the ontology concept instances are associated with IOs being developed and traced during the design episode. The usage of the proposed traceability methodology is discussed with examples of implementation and possible utilization situations. The paper is focused on explaining how the developed functionalities could help to resolve manifestations of inadequate information flow, which cause communication barriers in engineering companies. In addition, the proposed traceability methodology offers the possibility to record the detailed history of actions and events associated with IOs in the usage process of the product life cycle management systems. Based on the research findings, this paper argues that such a network of the traceability links and relationships may be viewed as a novel design communication channel.

Keywords: Collaborative Design; Design Communication; Engineering Information; Product Development Ontology; Shared Understanding; Traceability; Traceability of Engineering Information Project

1. INTRODUCTION

Communication is a multifaceted phenomenon that can be characterized in many different ways (Eckert et al., 2005). The ultimate goal of research on design communication is to improve the design process. Understanding how communication works and where it breaks down is an important step toward improving it (Eckert et al., 2005). The increased complexity of product development process, especially in large-scale projects, generates situations that may cause a communication breakdown. The impact of poor traceability practices on project efficiency may be considered one of these situations. A decrease in system quality, an increase in the number of changes, a loss of knowledge due to turnover, and erroneous decisions, misunderstandings, and miscommunication are some of the common problems that arise due to a lack of or an insufficient traceability of engineering information (Hurwitz & Kaufman, 2007).

Traceability of information provides the basis for assessing the credibility of engineering information, better understanding it, and making judgments about the appropriateness of its use for a particular design task (Storga et al., 2009*a*). In order to fully understand information and/or interpret various models and objects, it is necessary to know the context in which the data has been recorded. The stakeholders with different roles in the product life cycle process would like to have traceability carried out by traces of the product life cycle routes, because they want to reuse existing engineering information along sources, references, evaluation, meaning, reasons, arguments, documentation, choices, critique, and consequences (Storga et al., 2009*b*). They would like to leverage all relevant information regardless of its origin and format, and no matter where it resides, in order to help their organization innovate,

compete, provide service, and grow. The ability to trace development of engineering information becomes a prerequisite for better information value understanding and highlights the importance of communication quality in the product life cycle process.

We argue that an implementation of traceability in engineering design frameworks could significantly contribute to the quality of design communication. The aim of this paper is to present research results on how the proposed traceability architecture and methodology could help designers to improve communication and eventually create new channels of communication. This may be valid for all levels of communication interfaces: designer to designer, multidisciplinary team, team and company (organization), and interfaces of collaboration in an innovation network.

The discussion on facilitating design communication with traceability in this paper is based on empirical research, observations of industrial practice, and prototype software tool development and implementation done within a project on traceability of engineering information (TRENIN) research and development (Marjanovic et al., 2011*a*, 2011*b*). The objective of the TRENIN project was to develop a methodology for integrating and tracing the development of the information stored in diverse engineering information objects (IOs) during the product development process. Traceability may be defined as the ability to help stakeholders to understand the associations and dependencies that exist among entities created or used during a product development process. An overview of facilitating design communication through traceability is shown in Figure 1. The upper part of the picture represents all levels of design communication interfaces that may be supported with various aspects or areas of traceability. Two main directions of traceability may be distinguished:

1. *Looking forward—Guiding:* The traceability process is planned and organized, followed by assigning identification to IOs, activities, participants, locations, and resources, and exchanging it among participants. Here the participants should find the answers, for example, the overview of design process, the knowledge about information needs, the availability of information and documentation, and the relationships (linkages) between all identified items.
2. *Backtracking—Management of the design history:* The process should allow participants to follow the evolution of design items from its origins, through its development and specification, to its deployment and realization, and through periods of ongoing refinement and iteration in any of these phases. Tracing of the design history should improve understanding of the design routes by linking designed items to justifications, important decisions, and the assumptions behind them. By tracing designed items back to their sources, the impacts of later changes in any product feature can be identified before a product is redesigned.

The traceability framework includes predefined and recorded traces together with contextualization of all identified design items and participants in the design process. The ultimate goal is to make explicit and visible design items, the design process, and the information flow to all participants in the design process. The procedures that the traceability framework offers should act as enablers of design communication, as a kind of bypass around communication barriers, acting as a common place where missing information and knowledge could be found or be given. These new "indirect" channels of communication, together with enhancing the "visibility" of the design process and design items, could contribute to increasing shared understanding in distributed collaborative product development processes.

The ontology-based approach is used to define the context of information definition and exchange. Relationships between classes are used as a primary mechanism of linking the traceable items. Based on research findings gathered through observations conducted with TRENIN industrial partners, we expect that such an approach could lead to representations, interfaces, and visualization methods that will be easily understandable, efficient, and unobtrusive enough to be used and accepted in practice. The TRENIN project encompassed the development of several software tools, the most important of which is the interface to the product life cycle management (PLM) system, which automatically records the development (event) history of IOs that are managed in such a system (Storga, Marjanovic, et al., 2011).

This paper is focused on discussing the possibilities of improving (facilitating) design communication through the implementation of an ontology-based traceability framework and methodology. The discussion is guided by and attempts to answer the following research questions:

- Which factors that influence design communication could be impacted (improved) with traceability implementation?
- What are the ways of bridging the barriers and resolving the breakdowns in design communication with traceability?
- How could the traceability framework become one of the means (enablers) of creating shared understanding in collaborative design?
- How could the ontology-based approach lead to a traceability framework that will be efficient (and unobtrusive) enough to be used and accepted in practice?

The Introduction is followed by the literature review in Section 2, which aims to situate our work on traceability against the research filed of design communication. Relevant issues from theoretical and practical backgrounds are discussed in Section 3 to deepen the understanding of proposed traceability framework elements and structures. Those elements and structures were elaborated (Sections 4–6) based on studies and observations of current traceability practices
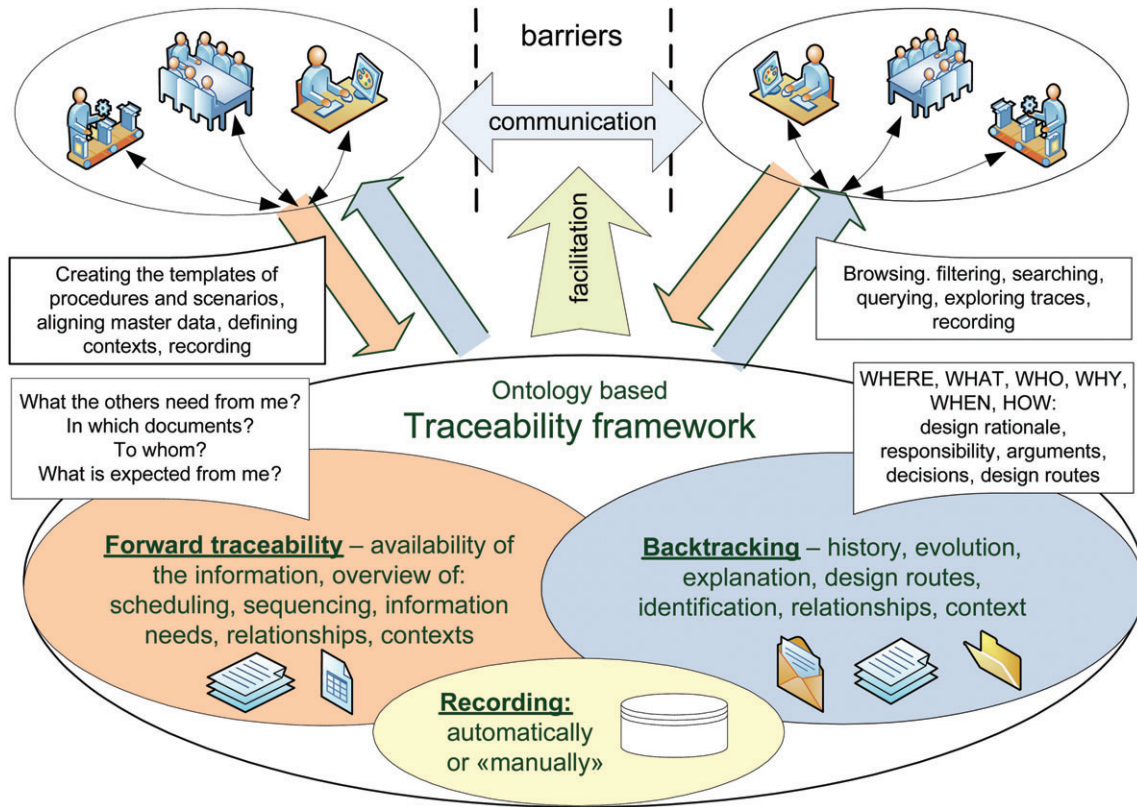
Fig. 1. The traceability framework as a facilitator of design communication. [A color version of this figure can be viewed online at http://journals.cambridge.org/aie]

with TRENIN industrial partners. The remainder of the paper presents an example of prototype implementation (Section 7) and finally an analysis of the possible contributions of the proposed methodology to design communication (Section 8). This analysis discusses proposed methodology with findings from Eckert et al. (2001), a paper that thoroughly describes manifestations of communication breakdown in large-scale engineering projects. Our conclusions are presented in Section 9.

2. LITERATURE REVIEW

2.1. Design communication and traceability

Eckert et al. (2001) reports observations of how failure to achieve appropriate information flow in large-scale engineering design processes contributes to a variety of problems for designers and decision makers. This paper includes a list of manifestations of inadequate information flow that have been observed in their studies. This list will be used as the basis for discussion in Section 8: how we believe that proposed traceability methodology and architecture could facilitate design communication. This discussion may be viewed as the main contribution and focus of this paper.

Flanagan et al. (2003) emphasize that individuals often have very little idea how their own tasks fit into the context of the wider product despite being experts in their own field

and understanding the tasks of the people they work with frequently. They may have little idea where information is coming from or going to and who is using it in the wider product context. Failure to exchange knowledge efficiently is often a symptom of communications problems. Based on an extensive industrial case study with a diesel engine company, Flanagan et al. (2007) argue that the overview and experience of senior designers play an important part in supporting teamwork by coordinating activities and facilitating proactive communication across large project teams. This paper proposes the use of design confidence, a metric that reflects the designer's belief in the maturity of a particular design parameter at a given point in the process. Confidence can be used to make overview knowledge explicit and to convey additional information about the design artifact, thereby informing communication and negotiation between teams.

A necessary condition for functioning communication is to understand the information need of the respective communication partner (Maier et al., 2009). The case study presented in Maier et al. observed the interface of design and simulation engineers at an automotive manufacturer.

A thorough review of the research on factors influencing communication and collaboration is presented in Maier et al. (2008). The authors explored the correlations between factors that influence communication in complex product development. Data for this analysis was acquired using the

"communication grid method" (Maier et. al., 2006). Insights into associations between factors that impact communication are seen as a lever to increase understanding and make the complexity of communication more transparent. Nine factors with a high linkage with each other were elicited:

1. availability of information about product specifications,
2. handling of technical conflicts,
3. roles and responsibilities,
4. mutual trust,
5. collaboration,
6. knowing what information the other party needs,
7. autonomy of task execution,
8. project reviews, and
9. overview of sequence of tasks in the design process.

Research done within the TRENIN project (Marjanovic et al., 2011*a*, 2011*b*) has shown that three of these factors are crucial for achieving traceability:

1. availability of information about product specification;,
2. knowing what information the other party needs, and
3. overview of sequence of tasks in the design process.

Moreover, mutual trust can be increased with traceability, because traceability provides the basis for assessing the credibility and maturity of engineering information and making judgments about the appropriateness of its use. We consider project reviews the crucial points in time where the most intensive recording activity has to be taken to achieve traceability.

Going back to the Figure 1, here we can conclude that for communication improvement it is more important to achieve (enable) "forward" traceability mechanisms to serve as enablers and to bridge the gaps (barriers) in communication. That means that actors in a design process would communicate easier and more efficiently if they know the following: what information they have to provide to the others, to whom each particular piece of information should be transferred, where and how the information about products and procedures is stored, what is the sequence of the tasks in the design process, and what level of confidence (trust) they could have in the particular information. Observations of traceability practice in the TRENIN industrial partners also yielded these factors (questions) as the most important constituents of traceability.

Research on factors influencing design communication continued in Maier et al. (2011). To improve 24 factors that have been empirically elicited in prior research, this paper collates more than a hundred recommendations from journal articles and textbooks published in the fields of engineering design, management science, sociology, and psychology. The contribution of Maier et al. (2011) is a list of recommendations for industry practitioners and an effort–benefit evaluation of individual recommendations.

2.2. Traceability and shared understanding in collaborative design

Kleinsmann and Valkenburg (2008) explored the barriers and enablers for the creation of shared understanding during a collaborative design process. Barriers and enablers are clustered according to their content: five clusters and a remainder category originated. All the clusters concern a different type of interface: an interface is an interaction pattern between actors. The authors found the following interfaces in the automotive case: between marketing and development, between the design team and suppliers, between the design team and the company, between the market researcher and the market, and between software development and the design team. One of the partners in the TRENIN project studies was also an automotive supplier, and there we also found that interfaces between marketing (sales) and development (the design team) and between the design team and company management are the main playground where the shared understanding has to be created. On the sales and development and the development and manufacturing interfaces we identified major "handover" situations (Eckert et al., 2005): scenarios in which a person undertakes a task, finishes it as far as possible, and then passes on the documentation to other specialists through a written specification.

Kleinsmann and Valkenburg (2008) constructed the factors that influence the creation of shared understanding between actors from different disciplines; here we would like to extract those factors that are connected with traceability or may be impacted by traceability:

- *on the actor level:* the ability of the actor to make a transition of knowledge, the view of the actor on the design task, the view of the actor on the process to follow, and the view of the actor of the knowledge to be shared;
- *on the project level:* the efficiency of information processing, the controllability of design changes, and the controllability of the project budget; and
- *on the company level:* allocation of tasks and responsibilities, and the availability of specialized knowledge within the company

According to Kleinsmann et al. (2010) knowledge integration is important in collaborative new product development. Kleinsmann et al. also claim that research in design communication finds that the quality of the collaborative new product development project is dependent on the process of creating a shared understanding. How could traceability contribute toward the creation of shared understanding? Traceability improves the availability of information and enables the creation of new paths (traces) for knowledge exchange and integration. Mohan and Ramesh (2007) define knowledge integration as the synthesis of specialized knowledge that is distributed across different artifacts and phases of product development into situation-specific systemic knowledge. Therefore the traceability tool acts as a platform that links various knowledge

chunks that are distributed across different stakeholders and their systems.

3. BACKGROUND AND RELATED WORK

3.1. Design rationale, design history, and knowledge reuse

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. Design rationale capturing tools are beginning to be accepted in industry, for example, the Design Rationale Editor (Bracewell et al., 2007). Agouridas and Simons (2008) argue that identification of latent or unarticulated customer and other stakeholder needs has been a significant barrier to improving the efficiency and effectiveness of the front-end phase of new product development processes. An in-depth determination of stakeholder needs entails analysis of their intentions; the overall aim of the work reported in this article is to establish a framework of intentional analysis and its associated methods and techniques to improve the traceability of design practice during the early phases of the design process. Shah et al. (1996) presented a database infrastructure for archiving and interrogating engineering design histories. They developed the design process representation language and the data definition language. Entities from the design representation language are similar (closely related) to classes constituting the ontology proposed in the TRENIN project.

Utilization, which is the reuse of recorded traceability data and knowledge, is really a complex area with many still-unclear issues and challenges. Hicks et al. (2002) emphasize that the reuse of knowledge is frustrated by semantics. The knowledge may not be structured and specific, and the particular application may require an altered perspective. Here the individual must dynamically step from knowledge back to information and then generate another perspective, which may provide knowledge for the new or unfamiliar situation. This new perspective is generated from a semantic interpretation of the old perspective. McMahon et al. (2002) introduce a user interface-based approach to the browsing of hierarchically organized information entities that avoids problems of word or phrase search.

3.2. Traceability difficulties in engineering design

Why is the achievement of engineering information traceability in modern, highly automated product development environments still so difficult? The current engineering design environments are not supportive of traceability procedures because people communicate and exchange engineering information across organizational and discipline boundaries, so they reuse existing information in new and unpredictable contexts and often information is translated from one format to another, during which information loss occurs. Furthermore, because of the lack of formal representations of the complex engineering design information, these exchanges still partly occur informally. As a consequence, retrieval of the engineering design IOs (e.g., with respect to format, type, and content) as well as correct interpretation of its content (due to the specific domain context) is difficult.

3.3. Advantages of the ontology-based approach

The proposed traceability architecture and methodology relies on product development ontology, which has the central role in information interpreting, defining context, establishing relations, and enabling indexing and searching mechanisms. In a description of utilization of their ontology-based model Brandt et al. (2008) say that semantic relations can be used to connect similar contents, and also to interrelate "content descriptions" with complementary contents, thus pointing to additional product knowledge that would otherwise remain undetected. Brandt et al. (2008) argue that ontologies have two major advantages over conventional data schemas: first, they are highly flexible, enabling modifications and extensions of the data structures even during project execution, and second, they are represented in a machine-readable, logic-based language, which allows them to formally define the semantics of the ontological concepts. As a consequence, it is possible to perform a semantic search on the ontology. Li et al. (2008) proposed a new computational framework that includes an ontological basis and algorithms to retrieve unstructured engineering documents while handling complex queries. The results from the preliminary test demonstrated that their method outperforms the traditional keyword-based search with respect to the standard information retrieval measurement. Storga et al. (2010) present the research of the nature, building, and practical role of a design ontology as a potential framework for the more efficient product development data, information and knowledge description, explanation, understanding, and reusing.

4. IOs: DEFINITION AND RELATED ISSUES

When talking about the traceability of information development or evolution, it is necessary to be clear about what it is that is being traced. While it might be desirable to do so, tracing the development of information per se is very difficult. It is not the information itself that lends itself to consideration but the physical or tangible manifestation of information. Chief among these manifestations of what are sometimes referred to as information as thing (MacLeod & Corlett, 2005) is the IO, or what is defined in the standardization community

as a record. The IO archetypically represents in the engineering domain a technical document that is often a complex structure constituted of information fragments, for example, textual and numerical elements and graphical representations of the technical content. According to the time period needed to create and complete the IO content, we may distinguish between IOs whose whole content is generated in a relatively short period of time and IOs whose creation takes a long time period in which the IO content evolves through numerous iterations and interactions with various participants in the design process.

Regarding versions, we may distinguish IOs having only one "stable" version from these IOs for which many versions are being generated in the IO life cycle. If a company uses a product data management/PLM system, then check in/check out procedures are the usual triggers for generating the new IO version. The main question here is what are the key differences between versions and, of the most importance, in which context?

If we consider responsibility for IO content from the level of the whole IO to the level of fragments, many variations may be distinguished:

- *One person* is responsible for the *whole content* of the IO in the *whole life cycle* of the IO;
- *One person* is responsible for *one or more fragments* of the IO in the *whole life cycle* of the IO;
- *Several persons* are responsible, each of them for *some fragments* of content in particular *time periods* of the IO life cycle; or
- *Several persons* are responsible for one fragment, which evolves.

We do not consider this list complete (fully comprehensive). In this view "persons" could be employees of company or they could be various company partners: outsourcers, suppliers, clients, service provides, and so on.

To achieve full traceability, it is obvious that it is necessary to develop IO fragmentation methodology. This, however, automatically raises new questions and challenges: Is it possible to determine a uniform fragmentation (for different viewpoints and contexts) for a particular class of IOs? We assume the answer is negative in most cases. Is it possible to always clearly distinguish fragments? Maybe this is possible only for strictly formalized and structured IOs. Criteria and context for IO fragmentation depend on the phase in the IO life cycle, who is using the IO content, and why. According to different fragmentation criteria and contexts, different fragments will occur. Some of the fragments may be the same for some criteria, but this will not necessarily always happen. Therefore, it is very debatable how to treat (manage) fragments of IO. Moreover, IO fragments originate and evolve in a sequence that is more or less difficult to predict. Mentioned challenges and issues of fragmentation were too complex to be tackled in this research phase; therefore, we decided to focus only on the IO level (as a whole), leaving fragmentation issues for future research.

5. OBSERVATIONS OF CURRENT TRACEABILITY PRACTICE IN INDUSTRY

Findings reported in this section draw on observations in two TRENIN industrial partner companies: an automotive industry supplier and a producer of energetic equipment and rail vehicles. A detailed and comprehensive report on these observations is given in Marjanovic et al. (2011*b*). The focus of these two studies was to analyze current traceability practices and communication problems. Typical for the automotive industry is that traceability has been brought to a very high level in the manufacturing process. Findings from our observations (as well as those of other authors, e.g., Ouertani et al., 2010) confirm that traceability in manufacturing and in the design process is really very different, with very little common implementation issues and problems. Therefore, we must emphasize that implementation of traceability in the design process is very labor intensive, requiring organizational processes and software support of a much higher complexity level than in the manufacturing process. In his PhD thesis, Smulders (2006) studied the changeover from product design to production. He emphasized that this is one of the transitions within the product innovation process that frequently causes problems and delays. The design is not ready on time, is not of the required quality, contains surprises for the production people, is too complex, or misses essential details, and so on. Smulders concluded that the individuals who participate in the innovation process and who intend to achieve a smooth transition and implementation are not sufficiently addressing all the possible adaptive issues that are necessary to realize their common goal. We believe that one of the ways to bridge the communication gap between design and manufacturing is to enable production engineers more valuable insights into design development history through backtracking (Fig. 1).

The main product of the studied automotive supplier company is a car seat, but it also produces many other components and subassemblies for about 30 car producers worldwide. Therefore, the negotiation process with buyers and suppliers has a crucial role for company success on the market. Current efforts in this company are primarily focused on further developing and implementing traceability in the process of negotiating with customers and secondarily on the design process. Responses in the negotiation process should be as quick as possible, so it is necessary to facilitate and accelerate the communication between all participants involved, but particularly important between sales engineers and company management. To further accelerate this process, traces to find information from previous similar negotiation situations (both successful and unsuccessful) could be of significant benefit. Availability of information, knowledge of information needs, and mutual trust here are the most important factors that influence communication in the following interfaces: team–company management, company–customers, and company–suppliers.

The analysis in both companies was based on interviews conducted with executive officers of main company departments,

members of management boards, and employees with significant responsibilities in the traceability process. In these interviews an emphasis was made on identifying problems in data flow and communication as well as identifying good current traceability practices. The further steps were to identify and analyze key documents, company internal analysis reports, and sets of rules and prescribed procedures impacting traceability (Marjanovic et al., 2011*b*).

Based on an analysis of the information and findings gathered, we have extracted major traceability issues and requirements relevant for project management in the automotive company:

- What are (were) the implications on the project management process for each transaction of one or more documents?
- Which documents are associated with one particular context or viewpoint?
- What is the completeness and accuracy of document content involved in a particular project milestone?
- Are all documents and information correctly and completely transferred from one main business process to another ("handover" scenarios between different teams)?
- What were the major business changes in the project portfolio, when and why did they happen, and how did they influence currently active projects?

Main traceability issues that are prescribed (in internal process organization standards) and recorded in a particular process are *who* provided particular inputs, *what* these inputs were, *who* did (executed) the activity, *what* the outputs of the activity were, and to *whom* the outputs were sent (proceeded). In this way, a subset of "traces" of activities and IOs enables all the process participants (who should perform tracing activities) to be aware of the traceability purpose. This process works in practice, but currently it is a mix of paperwork and computer-supported document management whose weaknesses are discussed in the rest of this section.

Traceability points and procedures that exist in current practice are mostly isolated islands managed by one particular person in each department. The automotive supplier company developed a concept of "monitoring sheets," which were created as spreadsheets or tables in a text processor. Chunks of data from these isolated islands are being transferred to a structure of big and complex spreadsheets, again managed by one responsible person: the executive officer of the document management department. This department is the central place for documentation management and approval, a place where actually the majority of traceability operations are performed. The document management department is also responsible for major handover scenarios between sales and development and development and production. The most complex traceability document structure being created is the project portfolio monitoring sheet, which traces major business changes and their implications to currently active design

projects. Lacking a common database and ontology, many redundancies and unnecessary data transfers are being generated. Such a huge structure of spreadsheets provides only very limited indexing and search mechanisms. There are several consequences: only one person could work with and access the file at a time, it is difficult to manage huge amounts of data, there are limited possibilities for avoiding and controlling errors, and only very simple structuring and capturing of newly created knowledge is possible.

6. TRACEABILITY METHODOLOGY AND FRAMEWORK ARCHITECTURE

6.1. Overview of the proposed traceability framework

Previously discussed issues directed further research efforts to the development of separate abstract levels for building the representation of IO relationships and putting them together around (in) specific contexts. In such a representation each IO could be repeatedly represented in various contexts. Various contexts may contain different subsets of IOs and their relationships. A concept of representation of such a context-driven representation of a subset of product documentation is named the "traceability record" (TR; Storga, Pavkovoc, et al., 2011). Besides context modeling, the main purpose of the TR is to be a container for recording the history (traces) of IO development through a specified time period called a design episode. Representation (definition) of a specific context is realized by extracting the subset of product development ontology developed as an essential part of the whole traceability framework. In other words, the TR identifies physical and abstract concepts and relations from the product development domain relevant for a description of the context of tracing a set of IOs in the product development process episode. Figure 2 shows the basic idea (concept) of TRs, which acts similarly to a semantic network in which IOs' relationships and tracing history are put together, indexed, and visualized.

A subset of the ontology on TRs is indicated here as a set of traceability element (TE) nodes. Every TE corresponds to one instance of one particular ontology class. When a particular design episode is finished, associated TRs are being stored in a database, enabling rich mechanisms for backtracking, interpretation, searching for information, and information origins. Here the IO is considered as defined in Section 4.

This basic description will be further developed to the level necessary to understand explanations of how the proposed traceability methodology could facilitate design communication. A complete description of a proposed system is out of the scope of this paper (it can be found in Storga, Marjanovic, et al., 2011; and Storga, Pavkovic, et al., 2011). TRs could also be defined as context-driven containers for recording traces of IO development. Context is defined by extracting the subset of product development ontology. Elements of the ontology subset are associated with IOs (in most cases, design documentation) belonging to the design episode that is to
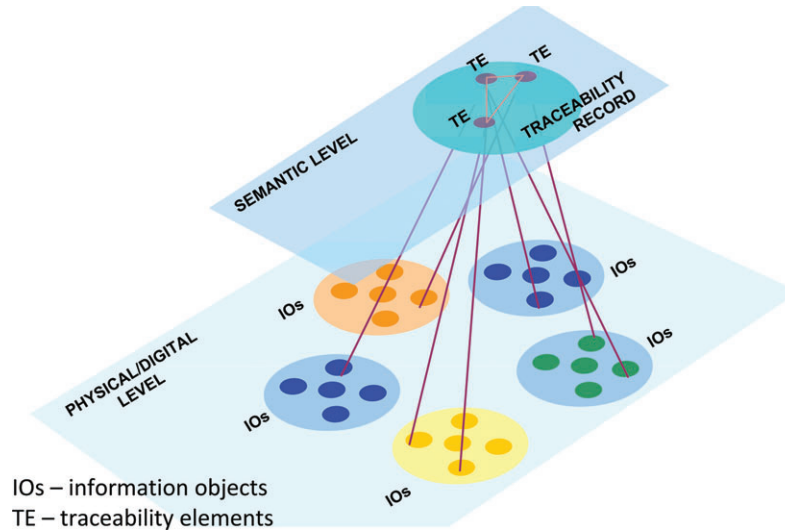
Fig. 2. The concept of traceability records. [A color version of this figure can be viewed online at http://journals.cambridge.org/aie]

be traced (Fig. 2, Fig. 3). A TR is a continuously updatable and accessible data structure that evolves parallel to referenced IOs being traced during specific product development episodes. The traceability episode covers a particular time interval and/or subprocess in the product development process, for example, a project phase or design task allocated to a team or individual designer. An episode is composed of a sequence of traceability events (TEVs) that represent key events and sets of activities and/or milestones in the product development process, usually prescribed within organizational and process workflow and modeled in existing software tools. TEVs are driving execution mechanisms of traceability episodes. TEVs may be initiated by external engineering applications (e.g., PLM actions like new, release, approve, update, delete), or they can be generated manually by the traceability

system users during the traceability episode. Part of system architecture named the traceability engine is responsible for recognizing that the associated event happened in a particular point of the traceability episode. In other words, each TR should be associated with the set of predefined events that trigger the process of capturing and recording traceability data and IO development.

Prior research (Storga et al., 2009b) suggests that a traceability framework should comprise a top-level ontology for the definition of TR templates. This allows flexibility either by customization or by extension. The set of TR templates can be customized and extended within the scope defined by the ontology according to the specific traceability needs. The idea is that the traceability manager in a particular organization selects relevant parts of the traceability ontology,
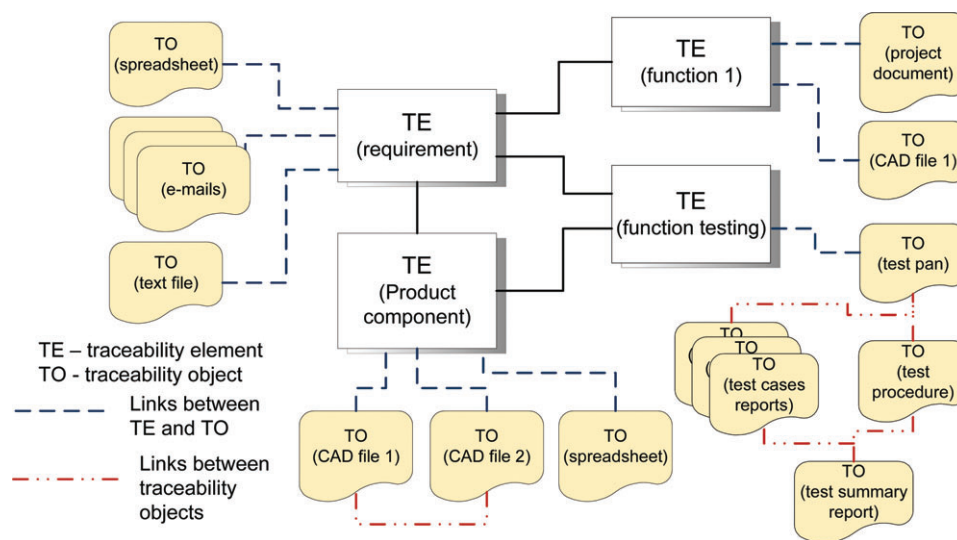
Fig. 3. An example of linking a traceability element and a traceability object on a traceability record. [A color version of this figure can be viewed online at http://journals.cambridge.org/aie]

expands them according to the traceability needs of a particular organization, and creates a set of TR templates that could be used in specific traceability episodes. In the presented research, we decided to adopt a merged ontology for engineering design (Ahmed & Storga, 2009), as a top-level ontology for the definition of the TRENIN formal language.

In the proposed TRENIN ontology two main areas have been distinguished: TEs and traceability objects (TOs). TEs represent (mainly abstract) concepts extracted from the design process domain, while TOs represent physical and digital entities from the design organization and management domain, with a special focus on design documentation previously defined as IOs.

From the perspective of object-oriented programming technology (used for building traceability framework), a TO is an object that represents a life cycle of the IO. In addition, regarding the representation of a particular technical document, a TO contains a data structure that holds the development history of that document and a set of other attributes necessary to achieve and implement traceability. Relations in proposed ontology are named traceability links. TEs are on TRs associated with the IOs managed by external applications represented with TOs, for example, project definitions, project plans, documents, items, users, flow processes, and so on. Relations between TOs should represent dependencies between content, hierarchy, timelines, and the like. As already discussed in Section 4, extending the model with a fragmentation of IO content is here crucial to achieving full traceability. TEVs are driving execution mechanisms of the tracing–recording procedure. TEVs represent events on IOs managed by an external application (e.g., new, release, approve, update,

check in, or check out) or are generated manually by the user during the traceability episode. Figure 3 is an example of one particular TR structure: each of four interrelated TEs has several associated TOs. Altogether these elements make up the particular context for tracing.

6.2. Traceability usage scenario

Considering how to achieve traceability in product development, we can emphasize the existence of the three main stages, whose description follows (Fig. 4).

1. *The identification and planning:* the main task is to define what (kind of) objects should be traced and what (kind of) links are needed between those objects. In other words, in this stage the TR is being defined and created. When starting a new development project, the designer chooses the particular TR template containing a set of predefined TEs, TOs, and the associated TEVs. The term *template* is used to emphasize that this is an initial data structure for building the complete TR of a development project. The most experienced designers could create TR templates for particular classes of development projects in a particular development office. In such a way, experienced designers could generate and disseminate the knowledge about how a particular project should be executed, monitored, and managed by their own experience. Previous research work, especially in design rationale capture (Kim et al., 2007), showed that designers usually are skeptical of such additional work. We argue that a template-based approach
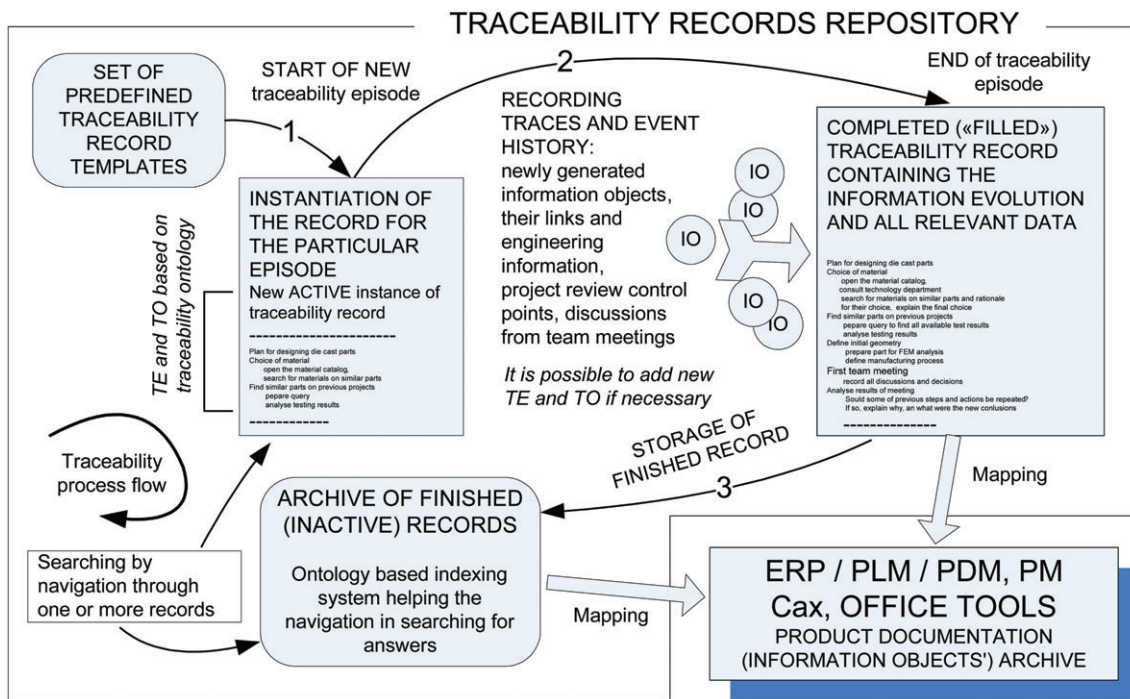
Fig. 4. The traceability usage scenario. [A color version of this figure can be viewed online at http://journals.cambridge.org/aie]

combined with the flexibility of an ontology extension would not require a significant amount of work for the derivation of new record structures once the main template structure is defined.

2. *Recording and documentation:* The main task of this stage is in the creation of traces that are the result of product development activities, designers' actions, decisions, reasoning, events, and so on. Those design items (IOs' origins and evolution) and design routes that have been explicitly defined in the TR should be recorded and documented for further use. Traceability operations according to associated events are being performed during all phases of product development. Through the timeline of development project realization, the TR is being updated, upgraded, and "filled" with traceability data (Fig. 4). The result of the development project episode is a finalized TR stored in a database.

3. *Using phase:* This phase has to provide understanding and reuse information in the 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's development issues or for other similar products;
- finding the source (origin) of mistakes or troubles; and
- simulating the design episode in another situation: for performing changes in existing solutions, reusing the existing solutions in new projects, configuration of the new variant of the product, and the educational process for inexperienced designers.

Answers to the above-listed issues could not be found in a straightforward manner. By choosing the appropriate active or stored records, the user starts the process of following, querying, and searching through recorded and documented traces. This process could be performed on the level of one TR or on a level of network of different related TRs. TR structural elements contain numerous links (references) to IOs that are being managed by various software tools. To achieve the satisfactory level of traceability, it is essential to keep the storage paths of archived IOs stable in time; otherwise, recorded references become useless.

6.3. Exploring visualization and utilization issues for initial implementation

One part of our research efforts was also directed toward exploring various methods and strategies for ontology-based indexing of predefined and recorded traceability items in order to find the balance between complexity and suitability for everyday engineering practice (Pavkovic et al., 2011). The research of Ahmed and Wallace (2003) identified two main advantages in having a visible indexing structure: to assist designers in focusing their query through browsing or navigating through indexing structures and to overcome difficulties in search engines not understanding the context of a query. In addition, we argue that following strategy is essential: when a need for tracing (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 with the initial indexing. Such an approach is important for better and easier understanding of search context and for providing an interface efficient and user friendly enough to be unobtrusive to designers overloaded with tasks and complex software tools. Previous research (Pavkovic et al., 2011) showed that it is very difficult to propose one common taxonomy/ontology structure that will equally suit the needs of all participants in a particular product development process. Following the methodology described in the previous section, the utilization interface and procedures are realized as set of reports and queries whose activation sequence will provide the user with a kind of browsing and "walking through" mechanism.

When a need for backtracking occurs, the search for relevant TRs begins with exploring (browsing) the same ontology structure that was used for TR creation. As a consequence, the database has to be searched for all the TRS that contain one or more instances of ontology concepts. This simple search should provide the user with the set of (or one particular) relevant TRs from which more complex utilization functions could be started. In the proposed approach, we have developed a report that statically shows the basic final structure of each finished TR (Fig. 5). This static structure is already a kind of filter where each element and object ID could be used directly for further searching and querying, or a kind of browsing. Figure 5 shows the basic structure of the proposed report. Starting points for browsing are menus activated by buttons placed near every element and object. This way the user could directly select the element or object to explore further information and/or relationships.

7. EXAMPLES OF INITIAL IMPLEMENTATION OF THE PROPOSED TRACEABILITY METHODOLOGY

The initial implementation and evaluation of the proposed TRENIN methodology is conducted in an environment of industrial partners. For this purpose, the TRENIN prototype software tool has been developed, comprising three main modules: the external application interface, the traceability engine, and the utilization explorer. The external application interface gathers data about life cycle events that occurred on IOs being traced. The prototype interface has been developed for a commercial PLM system being used in an industrial partner company. The traceability engine distributes gathered data to TRs and manages the process of recording traces. The prototype of the utilization explorer has been developed in a relational database using graphical query and report builder.

In this paper, a brief overview of one simplified implementation example is offered. For this purpose, a part of the devel-

Fig. 5. The database report view of the traceability record example. [A color version of this figure can be viewed online at http://journals. cambridge.org/aie]

opment process of a complex mechatronic product has been chosen: the vehicle control unit (VCU) for new generation of regional train. The VCU is responsible for controlling, measuring, sequencing, protecting, supervising, and communication tasks in the whole vehicle. Identified product development process structures, documentation, product hardware, and software components were mapped to proposed TEs and TOs. Reference models have been created to contain the company-specific instances of the ontology concepts; this way an extension of ontology has been developed, which is valid for this particular industrial partner. The traceability execution was simulated by the researcher following the particular development process.

The first implementation example is focused on the *early development phase: the consolidation of project and product requirements.* The main deliverable of this phase is the detailed product development plan. A detailed structured list of this TR is shown in Figure 5. The purpose of this record is to trace the creation and evolution of main project documents: the project development plan, the project Gantt chart, and the VCU technical data. TE "requirement gathering" is used to index several documents that are sources of requirements and other

data necessary in the creation of already mentioned main project documents. To complete the developed TR context, meeting minutes from the VCU development team initial meetings are also included, associated with element decision making. Figure 5 is also an example of basic database report view of one TR, which is a list of TEs and their associated TOs.

Each element and object has a button that activates further tracing possibilities and shows all basic data (authorship, creation date, etc.) for that particular object or element. For the selected TE the user could generate the list of all relationships with other elements on this TR, as well as relationships with other elements on other TRs. To start a broader search, the user could also get a list of all TRs containing the selected element. In addition, for the selected TO the system offers a list of event history, relationships with other objects, other records containing this object, all other objects of the same class, and so on. By combining the mentioned actions, the user is able to navigate through the search space of stored TRs and their contents, trying to find desired data or answers to complex questions.

The second implementation example traces *the process of hardware and software testing in the late development phase.*

Testing documentation comprises a huge set of documents with a complex hierarchical structure, as well as many other dependencies and data flows. 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 TRs. Testing results, especially test incident reports, are very interesting candidates to be backtracked in the utilization process. Let us describe one of the examples of likely utilization situations.

An electronic component from a previous project is reused in the current 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 the previous testing procedure for that component in a 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 a previous component usage and find an explanation of the causes of problem and solutions. If this component worked perfectly in the previous testing, at least the TRENIN user could collect all the data about environment conditions of previous component usage (interaction with other components, software version, operating system version, etc.), as well as data about persons responsible for component design. This could be a good starting point for seeking further details from other colleagues: now the designer knows whom to ask and what to ask, which should certainly facilitate design communication. Traceability frameworks could eliminate the need for unnecessary communication (especially in novice–expert interface) yet could direct the communication to correct channels, giving it the new dimension of quality. Such situations are broadly discussed in the next section.

8. DISCUSSION OF POSSIBILITIES THAT THE PROPOSED TRACEABILITY METHODOLOGY OFFERS TO FACILITATE DESIGN COMMUNICATION

The discussion in this section aims to answer how the proposed traceability methodology could facilitate design communication. Eckert et al. (2001) report on how failure to achieve appropriate information flow in large-scale engineering design processes contributes to a variety of problems for designers and decision makers. The main contribution of Eckert and colleagues' paper is a list that describes some manifestations of inadequate information flow that have been observed in their studies. This list is used here as the basis for discussion in the following way: each finding (relevant in context of traceability) and its explanation from Eckert et al. (2001) is cited and is then followed by our reflections on how the proposed traceability methodology might help (or might be upgraded) to resolve this issue or situation.

Lack of awareness of information history: Team members often don't know where items of information such as specifications and parameter values come from. Consequently, they can't trace them back to the designers who are responsible for them, and so can't question the information. If the designers need to change previous decisions such as parameter values, they don't know who has based their decisions on this value and therefore would need to change their own areas of the design. Tracking information is especially difficult across organizational barriers.

The proposed framework in its current state offers the possibility for backtracking on the document level, which is the first (easier) step to be achieved. Going into documents to the parameter level requires further development of the proposed traceability mechanisms. The other approach may be a major restructuring of design documentation: abandoning the classic file system in favor of the development of an integrated product and design process data model and structure. By integrated, we mean a data structure that is not dispersed through numerous tools, models, and not belonging to differently structured IOs. However, even with traceability only on the document (IO) level, designers could trace fragments of information (e.g., parameters) back to their origins, assuming that

- the author/owner of the document is also the person responsible for at least some fragments of information in the document, or
- the author/owner of the document knows where the particular information fragment came from.

Of course, there will be many cases and situations where one or both of these two assumptions will not be valid, but we believe that the proposed system offers the opportunities to find and identify paths and traces to and between information that could help designers to create an understanding of information history. At least in most cases it provides the information about who should be asked for further details. In our implementation example (Fig. 5), the content of requirements list document refers to set of documents also indexed by the TE requirement gathering.

Lack of awareness of how information is applied: Team members often don't know how their contribution fits into the overall process. They don't know who depends on the information that they are creating or how they use the information. Consequently, designers often don't provide their colleagues with all the information they need to have, especially about what decisions are provisional, or the boundaries within which parameters can be changed.

TRs comprise (gather) a subset of design documentation around a specific context. The ontology subset that defines the context also provides at least the partial explanation of one part or aspect of the overall process. How good these explanations (context definitions) would be significantly de-

pends on the overview of the whole process of the person who defines TE elements and relations between objects (see Flanagan et al., 2007). TRs should be structured in a way that offers designers a way to explore surroundings of their contribution, to see backward and forward at least through relations between ontology elements and associated IOs. To further improve the overview of fitting the partial contribution into an overall process, TRs should be hierarchically structured, enabling and showing connections and data flows through different operational levels. The proposed ontology includes both process- and product-related elements, therefore enabling navigation in process space and in product structure space.

> *Lack of awareness of changes to processes:* Design processes are often changed because new requirements are added or scheduled tasks have failed and need to be repeated or replaced with more complex procedures. News of process changes is often not passed on to members of the design team so that they don't replan their own activities or do tasks that are no longer required.

The proposed traceability framework is tightly connected with the PLM system, a part of the framework called the traceability engine recognizes and records all the events that happen in the life cycle of each document being managed in the PLM system (Storga, Marjanovic, et al., 2011). These events actually trigger the majority of the traceability operations in the proposed methodology. A detailed explanation of this process is outside the scope of this paper. Developed mechanisms of PLM event recognition might be upgraded toward developing procedures that will automatically pass the news of process changes to concerned users. According to the current state of development, this upgrade could be realized for a significant number (variety) of such situations, because we have developed a comprehensive procedure that distributes the news of process changes to all interested designers, for example, the announcement that a new IO is created.

> *No feedback on information provided:* Team members often don't get feedback on how their information has been used by colleagues. Consequently, they can't identify failings in how well they perform their own tasks or how they communicate with their colleagues, so they can't improve their performance. They may also feel underappreciated.

Improvements of communication in this situation require very complex tracing mechanisms; in other words, successors in the process have to evaluate the work of their predecessors and record this evaluation in the TR. This automatically raises the question of the objectivity of that evaluation: maybe this would be reasonable (feasible) only if experts evaluate the work of novices. Further observations and studies are required to develop this aspect (kind) of traceability.

> *No status information:* Team members can often not make sense of the status of the information that they receive, for example, if a parameter value is a final value or a simple estimate. People therefore often assume that values are exact and put great effort into meeting a seemingly exact target.

To resolve this issue, recorded traces have to include maturity status information, which is currently offered only on the level of the IO. Together with resolving the fragmentation issues discussed in Section 4, we should include the model of managing parameter maturity value status, for example, the sign-posting model developed by Clarkson and Hamilton (2000) and further discussed in Flanagan et al. (2003).

> *Power structure excludes viewpoints:* Contractors and suppliers are often excluded from decision-making processes because they have no official standing in the company hierarchy or because the information discussed in meetings is considered confidential. Yet their tasks depend on decisions made in these meetings; moreover, the success of the product might depend on the decisions made in these meetings drawing on their expertise and addressing their concerns.

Here we propose the approach of building the TRs exclusively for a specified class of users (e.g., suppliers), which will refer only those documents that those users need and may access. That way an automatic access-filtering mechanism is provided as well as an awareness of which documents and/or information are decided to be available for communication on external company interfaces.

9. CONCLUSIONS

The ability to trace the development of engineering information becomes a prerequisite for better information value understanding and acting on the importance of communication quality in the product life cycle process. The novelty of the proposed methodology is the usage of a subset of ontology to define the context for tracing and to act as a container in which the concept instances are associated with IOs belonging to the design episode that is to be traced. In such an approach, the selected ontology subset defines the semantically rich context for indexing and tracing a particular design episode.

The objective of developed utilization functionalities is to develop an approach to integrate and trace information fragments stored in diverse engineering working environments in order to improve communication in the product development process. In the discussion following the description of the proposed traceability methodology, we aimed to explain how the proposed traceability methodology and framework could facilitate design communication. The discussion was based on findings from observations of current traceability and communication practices in two engineering companies,

as well as on examples of the experimental implementation of the proposed system. We gave reflections on potentials and possibilities to bridge the gaps in information flows by implementing the proposed traceability procedures, particularly in large-scale engineering projects.

Besides bridging the gaps in information flow, the proposed traceability methodology offers the possibility to integrate knowledge toward the creation of shared understanding. Based on the proposed approach to defining context for tracing and indexing with TRs that are subsets of ontology, the knowledge integration could be accomplished in two ways:

- using the existing relations in ontology to navigate (perform semantic searches) between related elements of several TRs (which define different contexts), and
- establishing new relationships (either compositional or associative) between elements of TRs that will be recorded separately.

An example of tracing previous testing documentation, which is described in Section 7, illustrates one possible approach to knowledge integration and consequently avoiding unnecessary communication.

Experimental implementation led us to conclude that, when building the structure of TRs, it is very difficult to predict the needs for backtracking that will occur in the near future. This is valid particularly for complex traceability requests, for example,

- exploring the source (origin) of troubles that occurred in manufacturing or malfunctions in exploitation,
- finding relevant information and documentation to simulate an already executed design episode in another similar situation,
- using traces of previous design episodes for the educational process for inexperienced (novice) designers, and so on.

Therefore, in this phase of research the detailed overview of the proposed methodology utilization scenario(s) is still not completed. Further real implementations in a variety of companies would give us new insights and offer new ideas for necessary improvements toward abilities to answer more complex traceability requests. These experiments should be focused on comparing several approaches to answer which concept of building the TR could give users the best results in utilization and observing impacts on design communication: are there "visible" improvements in communication, and how are they manifested?

The TR concept also offers an approach for managing product and process information support by integrating information fragments from different IOs: an implementation of traceability on the fine-grained level. To achieve full traceability at the design parameter level, our further work will be directed toward developing mechanisms of IO fragmentation,

including parameter management methodologies (e.g., design structure matrix, MDM, and sign-posting). Such an approach should further contribute to two important factors that influence communication (already discussed in Section 2): an awareness of what information the other party needs and an overview of the sequence of tasks in the design process.

In the proposed utilization methodology, we have developed and demonstrated an approach that combines database reports, predefined queries, and filtering mechanisms that could be executed in any desired sequence. Furthermore, such an approach does not require knowledge of how to create and manage queries and reports. From the interface programming point of view, the novelty in the proposed model is the activation of querying procedures from a structured (grouped and sorted) report instead of the usual concept of using forms as interfaces. The basic report is a starting point for opening a variety of other reports that also offer further traces: this gives the user endless possibilities for walking through the network of traceability links. We argue that such a network may be viewed as a new kind of design communication channel.

REFERENCES

Agouridas, V., & Simons, P. (2008). Antecedence and consequence in design rationale systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 22*, 375–386.

Ahmed, S., & Storga, M. (2009). Merged ontology for engineering design: contrasting an empirical and a theoretical approach to develop engineering ontology. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing 23(4)*, 391–407.

Ahmed, S., & Wallace, K. (2003). Indexing design knowledge based upon descriptions of design processes. *Proc. Int. Conf. Eng. Design, ICED '03*, Stockholm.

Bracewell, R., Gourtovaia, M., Wallace, K., & Clarkson, J. (2007). Extending design rationale to capture an integrated design information space. *Proc. Int. Conf. Eng. Design, ICED '07*, Paris.

Brandt, S.C., Morbach, J., Miatidis, M., Theiÿen, M., Jarke, M., & Marquardt, W. (2008). An ontology-based approach to knowledge management in design processes. *Computers & Chemical Engineering 32 (1–2)*, 320–342.

Clarkson, P.J., & Hamilton, J.R. (2000). "Signposting," a parameter-driven task-based model of the design process. *Research in Engineering Design 12*, 18–38.

Eckert, C., Clarkson, P.J., & Stacey, M. (2001). Information flow in engineering companies: problems and their causes. *Proceedings of Int. Conf. on Eng. Design, ICED '01*, Glasgow.

Eckert, C.M., Maier, A.M., & McMahon, C. (2005). Communication in design. In *Design Process Improvement: A Review of Current Practice* (Clarkson, P.J., & Eckert, C.M., Eds.), pp. 232–261. London: Springer.

Flanagan, T.L., Eckert, C.M., & Clarkson, P.J. (2003). Parameter trails. *Proc. 14th Int. Conf. Eng. Design, ICED '03*, Stockholm.

Flanagan, T., Eckert, C., & Clarkson, P.J. (2007). Externalizing tacit overview knowledge: a model-based approach to supporting design teams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 21(3)*, 227–242.

Hicks, B.J., Culley, S.J., Allen, R.D., & Mullineux, G. (2002). A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design. *International Journal of Information Management 22*, 263–280.

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

Kim, S., Bracewell, R., & Wallace, K. (2007). Improving design reuse using context. *Int. Conf. Eng. Design, ICED '07*, Paris.

Kleinsmann, M., Buijs, J., & Valkenburg, R. (2010). Understanding the complexity of knowledge integration in collaborative new product develop-

ment teams: a case study. *Journal of Engineering and Technology Management 27*, 20–32.

Kleinsmann, M., & Valkenburg, R. (2008). Barriers and enablers for creating shared understanding in co-design projects. *Design Studies 29(4)*, 369–386.

Li, Z., Raskin, V., & Ramani, K. (2008). Developing engineering ontology for information retrieval. *Journal of Computing and Information Science in Engineering 8(1)*, 1–13.

MacLeod, R.A., & Corlett, J. (Eds.). (2005). *Information Sources in Engineering* (4th ed.). München: SK.G. aur.

Maier, A.M., Dönmez, D., Hepperle, C., Kreimeyer, M., Lindemann, U., & Clarkson, P.J. (2011). Improving communication in design: recommendations from the literature. *Proc. Int. Conf. Eng. Design, ICED '11,* Copenhagen.

Maier, A.M., Eckert, C.M., & Clarkson, P.J. (2006). Identifying requirements for communication support: a maturity grid-inspired approach. *Expert Systems With Applications 31(4)*, 663–672.

Maier, A.M., Kreimeyer, M., Hepperle, C., Eckert, C.M., Lindeman, U., & Clarkson, P.J. (2008). Exploration of correlations between factors influencing communication in complex product development. *Concurrent Engineering: Research and Applications 16*, 37–59.

Maier, A.M., Kreimeyer, M., Lindeman, U., & Clarkson, P.J. (2009). Reflecting communication: a key factor for successful collaboration between embodiment design and simulation. *Journal of Engineering Design 20(3)*, 265–287.

Marjanovic, D., Storga, M., Bojčetic, N., Pavkovic, N., & Stankovic, T. (2011*a*). *EUREKA E4911 TRENIN Project Final Report*. Accessed at www.trenin.org

Marjanovic, D., Storga, M., Bojčetic, N., Pavkovic, N., & Stankovic, T. (2011*b*). *EUREKA E4911 TRENIN Report on Stakeholder Perspectives on Traceability*. Accessed at www.trenin.org

McMahon, C., Crossland, R., Lowe, A., Shah, T., Williams, J.S., & Culley, S. (2002). No zero-match browsing of hierarchically categorized information entities. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 16(3)*, 243–257.

Mohan, K., & Ramesh, B. (2007). Traceability-based knowledge integration in group decision and negotiation activities. *Decision Support System 43*, 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.

Ouertani, M.Z., Baina, S., Gzara, L., & Morel, G. (2010). Traceability and management of dispersed product knowledge during design and manufacturing. *Computer-Aided Design 43(5)*, 546–562.

Pavkovic, N., Bojcetic, N., Franic, L., & Marjanovic, D. (2011). Case studies to explore indexing issues in product design traceability framework. *Proc. Int. Conf. Eng. Design 6, ICED '11,* pp. 131–140, Copenhagen.

Shah, J.J., Jeon, D.K., Urban, S.D., Bliznakov, P., & Rogers, M. (1996). Database infrastructure for supporting engineering design histories. *Computer-Aided Design 28(6)*, 347–360.

Shipman, F. M., & McCall, R. J. (1997). Integrating different perspectives on design rationale: supporting the emergence of design rationale from design communication. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 11*, 141–154.

Smulders, F. (2006). *Get synchronized! Bridging the gap between design & volume production*. PhD Thesis. University of Delft.

Storga, M. (2004). Traceability in product development. *Proc. Int. Design Conf., DESIGN 2004 2*, pp. 911–918, Dubrovnik.

Storga, M., Andreasen, M.M., & Marjanovic, D. (2010). The design ontology: foundation for the design knowledge exchange and management. *Journal of Engineering Design 21(4)*, 427–454.

Storga, M., Darlington, M., Culley, S.J., & Marjanovic, D. (2009*a*). Traceability of the development of "information objects" in the engineering design process. *Proc. 2nd Int. Conf. Research Design, ICoRD '09*, Bangalore, India.

Storga, M., Darlington, M., Culley, S.J., & Marjanovic, D. (2009*b*). Toward a process and method for tracing the development of information objects used in engineering design. *Proc. Int. Conf. Eng. Design, ICED '09*, Stanford, CA.

Storga, M., Marjanovic, D., & Savsek, T. (2011). Reference model for traceability records implementation in engineering design environment. *Proc. Int. Conf. Eng. Design 6, ICED '11*, pp. 173–182, Copenhagen.

Storga, M., Pavkovic, N., Bojčetic, N., & Stankovic, T. (2011). Traceability of engineering information development in PLM framework. *Proc. 8th Int. Conf. Product Life cycle Management, PLM'11*, Technical University of Eindhoven.

Neven Pavković is an Associate Professor in the Department of Design, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, where he received his PhD. He has gathered experience in industry working for major manufacturers of energy equipment in Croatia. Dr. Pavković has published over 40 journal and conference papers and has served on the scientific boards of two journals and numerous international conferences. His research interests include design process modeling and improvement, knowledge indexing and management, and design rationale capturing.

Mario Štorga is an Associate Professor in the Department of Design, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, where he received his PhD. He has been a Visiting Researcher at the Technical University of Denmark, Ecole Centrale de Paris, and the University of Bath. Dr. Štorga has published over 40 journal and conference papers and has served on the scientific boards of two journals and numerous international conferences. His research interests include PLM, product and process formal modeling, and design knowledge and information management.

Nenad Bojčetić is an Associate Professor in the Department of Design, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb. Dr. Bojčetić has published over 30 journal and conference papers and has served on the scientific boards of numerous international conferences. His research interests include computer-aided drafting, PLM, and knowledge management.

Dorian Marjanović is a Professor of product development and design in the Department of Design, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, where he received his PhD. His industrial experience includes quality control and design in heat exchangers production. Dr. Marjanović has published over 60 journal and conference papers. His research interests include product development theory and methodology, product development processes, product development and design management, knowledge and collaboration, design methods, and support tools for product development. He has been involved in a number of industrial and National Science Foundation research and development projects.