Traceability of Engineering Information Development in PLM Framework

Mario Štorga, Nenad Bojčetić, Neven Pavković, Tino Stanković

University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture
Ivana Lučića 5, 10000 Zagreb, CROATIA
Tel. 00385 1 6168 432, Fax. 00385 1 6168 286
Email: mario.storga@fsb.hr

Abstract: The work reported here builds on the framework for engineering information development traceability by discussing the strategy for traceability implementation within product life cycle management (PLM) environment. The four key processes in the complex product development practice (requirement-, change-, characteristic-, and decision management) have been considered in more details as a basis for the development of the approaches for traceability implementation in PLM environment. The traceability records with a goal to integrate process and product information that is fragmented across different information objects managed by PLM environments have been selected as backbone for implementation. Two possible approaches, scenario- and agent- based traceability have been proposed and evaluated. Research and development questions for the further steps in TRENIN (www.trenin.org) project progression have been identified and described.

Keyword: Engineering information, product lifecycle management, scenario based approach, agent based system, TRENIN

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 [1]. 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 [2]. As the product lifecycle proceeds, engineering information will be used to document the decisions taken, describe potential limitations of existing solutions or their suitability for adaptation, and to identify what further information is needed. Throughout this process, the information will be evaluated and recorded by members of the product lifecycle team in a variety of formats, such as sketches, drawings, notes and meeting minutes. In order to support the product lifecycle as it progresses, a proportion of this information will be formally recorded in technical reports and other engineering documentation, such as CAD models, production drawings, calculations, installation instructions, etc. It can therefore be argued that the efficiency of the product lifecycle process is highly dependent on the effective utilisation of this existing engineering information.

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 [3]. Traceability has been considered as a quality attribute and many standards governing systems development require the creation of traceability procedures. 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. 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 [4]. The work reported here builds on the **TR**aceability of **EN**gineering **IN**formation - TRENIN (www.trenin.org) framework for engineering information development traceability by discussing the strategy for traceability implementation within product life cycle management (PLM) environment.

2 PLM state of the art

As Stark [5] postulates, PLM is the activity of managing a company's products and information about products across the complete lifecycle, from the early stages of conception to the final disposal or recycling of a product. As a comprehensive business tool, PLM involves the fusion of many traditional engineering disciplines such as computer aided design (CAD), computer aided manufacturing (CAM), and computer aided process planning (CAPP) with many traditional management disciplines such as lean manufacturing and six sigma quality control, supply chain management (SCM), and enterprise resource planning (ERP) [6].

The fusion is made possible by the rapid advances in computer, information and communication technologies. PLM in general, is today considered as an instrument for enabling the company to provide an additional value from their information to the customers and thereby gain a competitive advantage over their competitors [7]. PLM strategy is usually followed by information technology that allows faster, cheaper and better conception, invention, feasibility, testing and deployment of products. PLM allows significant improvements to the quality, cost, life, reliability and environmental implications of existing and new products. PLM allows seamless creation, training and deployment of products and information with embedded mechanical, chemical, electrical, computer, intelligence, and communications hardware and software.

The following list shows one way of classifying the functions of the PLM environment (Table 1): product data management (PDM), product and process definition, configuration management, customer-oriented collaboration, visualisation/viewing, data exchange, definition and management of the product lifecycle processes, project and portfolio management, system integration, etc.

Table 1 Functionality relevant for traceability issues that currently exists in engineering tools

PDM	CAD	OFFICE TOOLS
• Project management Document versioning management • Workflow mechanism • Engineering change management • Search/querying engine • Reports generator	• Feature tree (structure of the CAD model) • Associatively links between assemblies and parts • File versioning • Product 3D model characteristic management	• Track changes mechanism • Document properties management

Collaborative processes and technologies have dramatically improved the value of PLM systems that help companies better manage product information. But, in the same time many companies still suffer from diminished innovation and product development capabilities because of fragmented, disjointed information. The preset practice of recording the outcome of the product lifecycle process is for highly formalized model of the product to be produced, in the form of conceptual sketches, calculations, computer-aided engineering models, bills of materials, engineering change orders, maintenance instructions, etc. However, the detailed process, activities and rationale by which the product has been designed and created, and engineering information has been developed, are recorded in poorly accessible informal manner (if at all). Consequence of such practice is lack of engineering information origin understanding and danger of mistakes during existing information retrieval, adaptation and integration.

3 Traceability of engineering information development

The different stakeholders in product lifecycle process would like to have traceability carried by traces of the product lifecycle routes, because they want to reuse existing engineering information along sources, references, evaluation, meaning, reasons, arguments, documentation, choices, critique, consequences [4]. They would like to leverage all relevant information no matter where it originated, no matter what its format, and no matter where it resides in order to help their organization innovate, compete, provide service and grow. Ability to trace engineering information development becomes prerequisite for better information value understanding and recognition and act on the importance of information quality in product lifecycle process [8].

Little is currently understood about the requirements for engineering information traceability in product lifecycle and there are few methods by which effective traceability can be ensured. There are a number of methods and tools which contribute partially to the traceability of information development in general, but the emphasis here is either on description of the product data management (PDM) or project/workflow management rather than the explanation of development and rationale on information antecedents.

Traceability should assist in understanding the relationships that exist within and across product lifecycle information like requirements, design details, component description, production specification and maintaining procedures. These relationships help engineering designers to understand the rationale behind the design procedures during product development. The need for maintaining traces among information objects to support change management in product development is well documented in our previous publications [3], [4]. Literature also describes the adverse impact of poor traceability practices on project cost and schedule. Decrease in system quality, increase in the number of changes, loss of knowledge due to turnover, erroneous decisions, misunderstanding, and miscommunication are some of the common problems that arise due to lack of or insufficient traceability knowledge .

Traceability records should help in maintaining a semantic network in which nodes represent information objects among which traceability is established through links of different types and strengths. The simplest traceability tools that have been found in engineering practice during the interviews with our industrial partners are purely relational (i.e. in the form of spreadsheets or personal notes) and do not systematically distinguish different node and link types. They are suited only to support simple traceability practices for personal use and provide limited support for information dependency analysis. In our project, this lead to the development of a traceability records with a goal to integrate process and product information that is fragmented across different information objects managed by PDM/PLM environments.

4 Key processes to be supported by future traceability methods and tools

Little is currently understood about the engineering information evolution traceability and there are few methods by which effective traceability can be ensured [9]. Different research groups approach to the many parts of the traceability issue through perspective of knowledge integration [10], communication, handling complex dependencies between requirements and components [11], task-specific management [11], ontological retrieval of the unstructured documents [12], traceability schemes for integration of the product and process knowledge, appropriate information flow achievement [13] and architectures of the information search and retrieval systems [14].

In addition, there are no existing tools that support achievement of the full traceability of engineering information evolution in product development. Currently available PDM/PLM systems support information exchange between product developers, especially in the later phases of the engineering lifecycle which is characterized by more deterministic and well-known processes. However, they lack essential capabilities for the management and use of product information. Some recent research efforts try to extend the capabilities of PDM/PLM systems for product information traceability during the product development phase [15]. The key issue with the traditional traceability approach, in particular from the point of view of industrial applications, is that it is labour intensive, both for the product information-engineering specialists as well as for those whose information they are seeking to acquire. This PDM/PLM repository in practice is usually limited to the storage of product data and documents. It does not offer support for the recording and management of the associated work.

In order to recognise key issues for the traceability records specification, modelling and implementation, we have decided to consider in more details the four key processes in the complex product development practice as a basis for further development of the new approaches for traceability implementation in PLM environment.

4.1 Requirements traceability

Requirements are the subject of an extensive body of literature in the information systems domain. Some of the work from this domain has been investigated with a view to making recommendations for traceability of the requirements in engineering design [16]. The following definition sums up the general view of the requirements traceability [17]: "*The requirements traceability is the ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases.*". In requirements definition phase it is important that the rationales and sources of the requirements are captured in order to understand requirements evolution and verification. Modifications during design appear e.g. if the requirements evolve or if the product is developed incrementally. During design phase requirements traceability allows to keep track of what happens when change request is implemented before a product is redesigned. Traceability should also give information about the justifications, important decisions and assumptions behind requirements. Test procedures on prototypes can be traced to requirements or design and this kind of traceability helps to design and modify test procedures. Modifications after the delivery of the product will happen due to various reasons (e.g. to correct faults or to adapt the product to a changing environment).

4.2 Changes traceability

To implement an engineering change request, change management strategy helps to identify necessary changes and understand the impact of changes. In general, the objective of different change management practices is to ensure a systematic development process, so that at all times the system is in a well-defined state with accurate specifications and verifiable quality attributes. Change management helps in the management, control, and execution of change and evolution of product, while traceability usually helps in managing dependencies among related artefacts across and within different phases of the development lifecycle [18]. In vast majority of organizations, these two practices are implemented in isolation. The lack of knowledge about how the process and product information are related makes it difficult to understand different viewpoints held by various stakeholders involved in change process and estimate the impact of changes, thus hindering change management and adversely affecting the consistency and integrity of systems. Without the capability to acquire and trace engineering information development, it is very difficult to incorporate modifications in the system. Therefore, change management should not only help manage changes to products of development (product knowledge), but also help trace the effects of the changes on other information entities (dependencies) and the reasons behind such changes (e.g. rationale) to maintain consistency among the various information entities.

4.3 Characteristics traceability

The definition of key product characteristic is one of the gifts of automotive manufacturing to all other kind of production. It is quite impossible to cost-effectively measure every possible characteristic of a given product. However, it is possible to define the most significant characteristics as key product characteristics (KPC). For example, the front of an instrumentation cluster may have significant appearance requirements, but is usually not necessary that the back of the product (invisible to the operator) have the same level of appearance quality. Hence, the front appearance and its definition is a key product characteristic. A KPC is a feature of material, process or part where the variation within the specified tolerance has a significant effect on product fit, performance, service life or manufacturability [19]. A KPC should be identified only after determining a significant benefit exists from controlling the characteristic to assure that the feature is at or very close to the specified value. KPC is usually identified as a part of the product development. Once a KPC has been identified, variation management activities must be performed until the process or processes that influence that characteristic are in control and process capability has been established. Appropriate traceability methodology for the key product characteristics should be implemented to assure continued performance of the products life cycle process.

4.4 Decisions traceability

In complex group decision and negotiation (GDN) activities, the participants access and use information about the problem and solution domains, which is stored in a variety of information sources such as spreadsheets, meeting minutes, design documents, etc. Seamlessly linking such information fragments spread across organizational work processes and tools will be very helpful in supporting GDN activities [20]. Creation of such networks by seamless integration has been attempted by many tools handling explicit, codifiable content (e.g., workflow tools, project management systems, collaborative systems, intranets, and data warehouses) and those that enable sharing and distribution of contextualized information content (e.g., digital whiteboards, case-based reasoning tools, multimedia channels, annotation tools, and concept mapping systems). One of the common problems in facilitating integration of information objects to support collaborative product development is that the stakeholders involved do not have adequate guidance on what kind of information elements should be integrated, and how the integration should be structured and used. Traceability, defined as the ability to describe and follow the life of a physical or conceptual thing, addresses these challenges by providing semantic and structural guidance to information objects integration. We could argue that integrating information fragments used by various stakeholders by providing traceability among them will increase the effectiveness of GDN activities performed during the product development process. Information objects traceability network can be defined as a semantic network in which nodes represent different information objects among which traceability is established through links of different types. Such a network facilitates the understanding and communication of the context in which group decisions and negotiations are carried out and help in monitoring the repercussions of changes in the underlying context.

5 Traceability implementation approaches

Based on the extended literature overview and discussion with research and industrial partners regarding the support that is expected from traceability implementation in engineering working environment, two possible approaches have been recognised:

- User predefines what exactly and how would like to trace in particular episode – **Scenario Based Traceability** approach (**SBT**).
- System is automatically or semi automatically tracing everything related to the information objects life continuum and enable users intelligent search among this records – **Agent Based Traceability** approach (**ABT**).

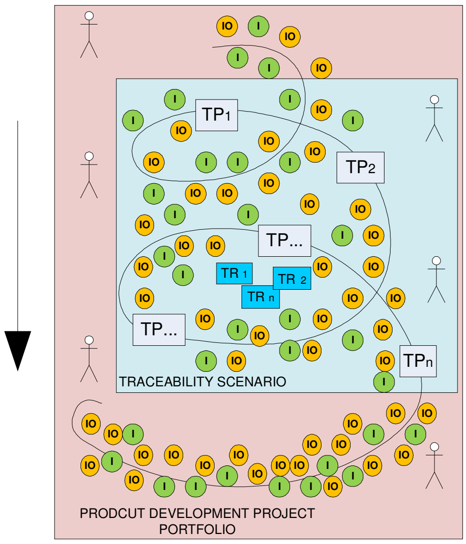

Figure 1 Scenario based traceability (SBT)

Scenario Based Traceability (SBT) approach is developed on some presumptions about information objects in product life cycle. Information object are characterised and described by different attributes like format, purpose, life continuum availability, content, form, versioning, status, responsibility, source, identification, fragmentation, links, etc. Traceability scenarios therefore should cover specific time interval in space-time continuum of the product lifecycle process. In that continuum, scenarios define set of the traceability points (TP) representing key events important for traceability of engineering information development. For each TP, scenario should define the structure of the traceability records (TR) that maps state change for the key engineering information that should be captured by proposed scenario (Figure 1).

Traceability record is defined as a record of the information objects changes and development including attributes, links and procedures that controls TR in particular TP. Traceability record is imagined as a „glue" for the information content that it maps. Examples of the state change that could be recorded as a TR are: initialisation, use of content, semantic relation of the information objects and their fragments to other information objects, creation of the traceability record, etc.

Agent Based Traceability (ABT) approach was built around idea of extending existing PLM environment with intelligent agent technology in order to enable autonomous traceability actions necessary for traceability execution. The main ABT schema is presented on Figure 2. The core of the idea is traceability engine (TE) that, based on the specific events related to the PLM environment and PLM information objects, executes "intelligent" agents responsible for traceability tasks related to specific event. Agent management is done by agency is responsible to select right agent from the agents' pool and based on the description of necessary traceability behaviour executes it. That sequence result with traceability record in database. The main idea behind this approach is that current state of the PLM information object is a superposition of initial state and changes over the time. Therefore, the ABT traceability table will contain records of the every change of the PLM information object (including content, attributes, links, etc) and what is especially important, context of the change provided automatically or by help of human user involved in traceability process. The meta-data is a meaning for specification of the information or information fragments that will be recorded by ABT.

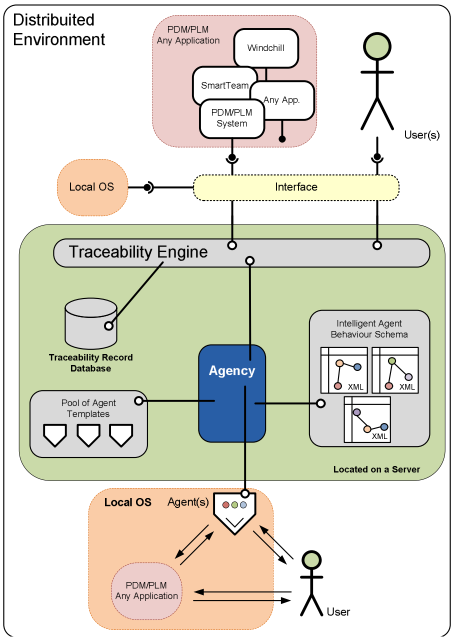

Figure 2 Agent based traceability (ABT)

After further development of the two approaches and discussion with potential users from industrial practice, the concepts like traceability record, traceability object, traceability point, and traceability engine have been clarified. The main advantage of the SBT approach is contextual richness while ABT records more information that could be, in a perspective, base for advanced automatic reasoning on traceability routes. The main problem with SBT is need to predefine all the possible traceability scenarios that could be of interest for different stakeholders in product lifecycle process, limitation on richness of available information and lot of manual work and interaction by the user. The problem of the ABT is formalism of the engineering information that should be fully respected in order to implement intelligent agent system, scalability of the potential and semantically rigidness needed. It was also recognised that TP in SBT could suit to events in ABT. TR that was imagined as a static list in SBT could be seen as a simplification of the agent and ABT traceability table that is more dynamic. SBT is limited by start/end time moment, and ABT by baseline in PLM.

Even though the two approaches have shown differences, the team finally concluded that would be smart in the further development to consider how two approaches could benefit from each other and be merged in single TRENIN architecture proposal. The key decision for the further development could be summarised as follows:

- **TRACEABILITY POINT** could be seen as an external EVENT that should be related to the product development process like workflow in PLM, and not only to the information life continuum activities. Research and development question that arise from this decision is about granulation of the engineering process and engineering activities that should be considered.
- **TRACEABILITY RECORD** instead being the pure static list of the information objects and hyperlinks between them should be more “intelligent” and dynamic container of the traceability elements, information and links semantically enriched in order to provide the context of the informational content development. The research and development question that should be answered in further research and development is about structure and properties of the smart traceability records.
- **TRACEABILITY ENGINE** should extend pure records of the information objects’ state increment with context of the changes in order to engineering information be more useful for understanding and reusing. The research and development question from this conclusion is about development of the vocabulary or ontology for the information objects development context description.

- **TRACEABILTY FRAMEWORK** should be implemented independently from PLM system since it has to be integrated with different types of the document management, file management, engineering data management and product data management systems that are currently used in engineering environments. The research and development question that should be answered in further development is about architecture of the traceability framework and integration with PLM functionality.

6 Conclusion

Consideration of the strategy and possibilities for the traceability implementation framework within product life cycle management (PLM) environment, has led closer to fully specified TRENIN implementation architecture. The following progress has been made:

- Identification of the shortcomings related to traceability functionalities in existing engineering tools, with focus on PLM systems.
- Key processes related to the complexity of the product development context to be supported by future traceability methods and tools have been identified and explained in cooperation with industrial partners.
- Two different traceability implementation approaches have been proposed and evaluated.
- Research and development questions for the further steps in project progression have been identified and described.

It is expected that further implementation of the TRENIN models and methods in PLM environment will enable semantic and structural guidance to full engineering information objects integration and smarter utilisation during product life cycle.

Acknowledgment

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

1 McAlpine H., Hicks B.J., Huetand G., Culley S.J. (2006) `An investigation into the use and content of the engineer's logbook`, *Design Studies* 27, Elsevier Ltd., pp. 481-504

2 MacLeod R.A., Corlett J. (eds.) (2005) `*Information sources in engineering*` (4th ed.), K.G. Saur, Munich, Germany

3 Štorga M., Darlington M., Culley S., Marjanović D. (2009) `Traceability of the development of 'information objects in the engineering design process`, *Research into Design - Supporting Multiple facets of Product Development* / Chakrabarti, Amaresh (ed.), ICoRD'09 Bangalore, Research Publishing, India, pp. 263-270

4 Štorga M., Darlington M., Culley S., Marjanović D. (2009) `Toward a process and method for tracing the development of information objects used in engineering design`, *Proceedings of the 17th International Conference on Engineering Design – ICED'09*, Stanford, The Design Society, USA

5 Stark J. (2004) `*Product Lifecycle Management: 21st Century Paradigm for Product Realisation*`, Springer USA

6 *https://engineering.purdue.edu/PLM/*

7 Golovatchev J.D., Budde O. (2007) `A holistic Product Lifecycle Management framework facing the challenges of 21st century`, *Proceeding of the 3rd International Conference on Life Cycle Management - LCM2007*, Zurich, Switzerland

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

9 Ouertani M.Z., Baina S., Gzara L., Morel G. (2010) `Traceability and management of dispersed product knowledge during design and manufacturing`, *Computer-Aided Design*, Elsevier, *in press*

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

11 Gausemeier J., Dorociak R., Pook S., Nyßen A., Terfloth A. (2010) `Computer-aided cross-domain modelling of mechatronic systems`, *Proceedings of International Design Conference - DESIGN 2010*, The Design Society, FSB, pp. 723-732

12 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, pp. 263–280

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

14 McMahon C., Lowe A., Culley S., Corderoy M, Crossland R., Shah T., Stewart D. (2004) `Waypoint: An Integrated Search and Retrieval System for Engineering Documents`, *Journal of Computing and Information Science in Engineering*,

15 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 2008*, 32 (1-2), pp. 320-42

16 Baxter D., Gao J., Case K., Harding J., Young B., Cochrane S., Dani S. (2008) `A framework to integrate design knowledge reuse and requirements management in engineering design`, *Robotics and Computer-Integrated Manufacturing*, Volume 24, Issue4, pp. 585-593

17 Ramesh, B., Jarke, M. (2001) `Toward Reference Models for Requirements Traceability`, *IEEE Transactions on Software Engineering*, Vol. 27, No. 1, pp. 58-93

18 Mohan K., Xu P., Cao L., Ramesh B. (2008) `Improving change management in software development: Integrating traceability and software configuration management`, *Decision Support Systems*, Elsevier, Volume 4,5 pp. 922-936, 2008.

19 Barker G. (2003) `Critical characteristic and key product characteristic`, *The American Society of Quality, The Aviation/Space and Defence Division newsletter,* Vol. 35 Number 2

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