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

TRACEABILITY VISUALISATION TOOLKIT

T. Stanković, M. Štorga, I. Stojić and T. Savšek

Keywords: traceability, evolution, information, visualisation, design, TRENIN

1. Introduction

The importance of engineering information is underlined by the fact that product lifecycle as chain of information transformation processes both consumes and creates large amounts of information [McAlpine et al. 2006]. As the lifecycle proceeds, the information originating from various sources will be captured, interpreted, evaluated, recorded and reused by stakeholders involved in product lifecycle. A proportion of this information will be formally recorded in various media and form as technical reports, meeting minutes or CAD models to ensure reusability and adaptability for future utilisation. How well it may be to meet the challenges of complex products and services and their lifecycles is highly dependent on the effective utilisation of this existing engineering information. One way by which the utilisation of information can be fostered through is by applying various computer supported visualisations. The latter assumes a number of techniques which are utilised to provide processing, comprehension, and retention of information in static, animated, dynamic, and interactive graphics [Plötzner and Lowe 2004]. User friendly graphical interfaces with underlining algorithms, through which information can be visualised as spatially organized and interactive, alleviate the information understanding and retrieval process thus supporting problem solving that engineers confront today. The work reported here builds on the TRaceability of ENgineering INformation - TRENIN (www.trenin.org) research project [Štorga et al. 2011[a]], [Štorga et al. 2011[b]] by discussing the approach to visualisation of the engineering information evolution during product development.

Traceability of engineering information could be viewed as one quality criterion of a product development environment with the main goal being to ensure that engineering information is clearly linked to its background, origins, rationale and sources during its development. By providing a technology for tracing the development of the engineering information, a means is provided for making the product and processes during product life fully traceable [Štorga et al. 2009]. Research in TRENIN project lead to development of *Traceability Records* (TR) to integrate process and product information that are fragmented across different information objects managed by PDM/PLM environments [Štorga et al. 2011[b]]. *Traceability Records* are engaged in maintaining of semantic network in which nodes represent information objects among which traceability is established through semantic links of different types and strengths. Understanding the relationships which emerge across engineering information in product lifecycle like requirements, design details, component description, production specification, maintaining procedures and key product characteristics demand a selection of suitable visualization technique to convey the relevant meaning expediently. To interpret a fragment of information in product life cycle in respect to the rationale behind it, it is necessary to understand the circumstances, namely to grasp the time dependant context in which the information has been developed, recorded, updated or retired (Figure 1). The visualisation is hereby called to display dynamical relationship between emerging information as input by many sources during product life cycle.

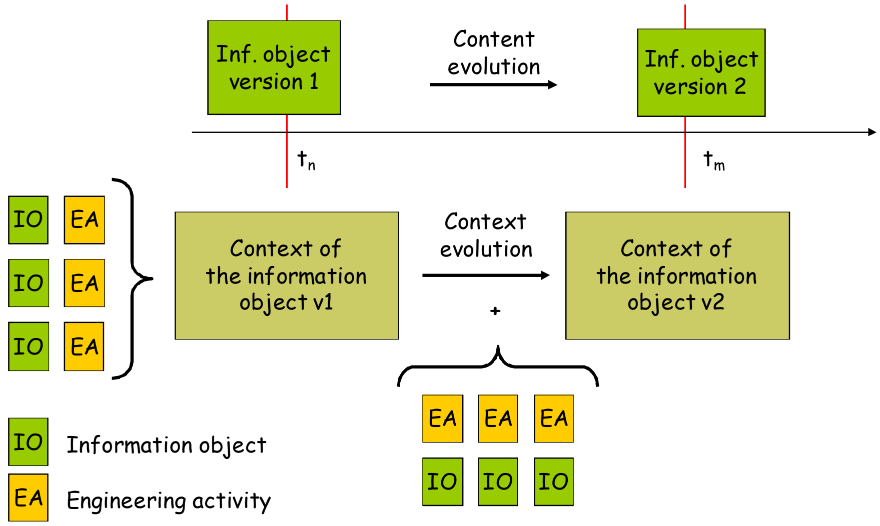

Figure 1. Evolution of engineering information content and context [Štorga et al. 2009]

In order to improve utilization phase and provide a context by which the engineering information can be better interpreted, one of the goals of the project was to develop visualisation technique for the *Traceability Records* which should enable more effective navigation and search, provide better understanding and put the engineering information in proper context in which it is supposed to be managed and reused.

In general, data visualisation techniques are complementary to statistical data analysis; as such they are applied for refining and exploring data or for creating results presentations [Bolton et al. 2004], [Unwin et al. 2006]. Usual practice found in industry of representing engineering data and information in a form of table based output summaries as spreadsheets can suffice simple traceability practices which are suitable for personal use providing limited support for information dependency analysis. However, even for such cases when information builds up after a while, it is difficult to perform a systematic search to distinguish the information within not being able to find what has been searched for. What is tried to be pointed out here in respect to the visualisation of information is maybe best defined [Card et al. 1999] as the use of computer-supported, interactive, visual representations of data to amplify cognition. The effective utilisation of existing engineering information is thus not only dependant on the information recording and structuring framework but also by the way information is represented and managed by the user. A visualization technique suitable for utilisation of engineering information has to offer some kind of narrative explanatory approach [Fry 2008] by which user can refine and navigate the information and context by intent to uncover the meaning behind the records. By making information structures organized, modern visualisations provide means for user to interactively navigate and uncover the information engineers are looking for [Keller and Tergan 2005]. It is presumed that the user is often being unaware of the precise information location by which the information can be obtained or possesses incomplete specification relating the information necessary to perform search. Both of the latter could be the cases in the product development of the complex technical systems involving large data and information sets and multitude of stakeholders generating and interpreting information.

The purpose of this paper is to present organic approach for visualisation of engineering information evolution during product development. From the literature [Fry 2000] an organic information visualisation assumes a system which employs simulated organic properties in an interactive, visually refined environment to collect qualitative facts from large bodies of quantitative data generated by dynamic information sources.

The next section will address organic information visualisation in more detail. Third section will deal with an overview of visualisation tools and environments which we have considered for the TRENIN visualisation. Relating engineering information traceability and evolution visualisation is addressed in the Section 4. Case study and discussion are presented in following sections. Conclusions and further work will close this paper.

2. Background on organic information visualisation

Organic information visualisation follows a systemic approach to information visualisation. Built on the mimicking of behaviour and properties of living (organic) systems, organic visualisation aims to augment one's ability to process and understand complex data [Fry 2000]. Organic information visualisation is complementary with the usual parse-filter-mine-represent-refine-interact data mining scheme [Unwin et al. 2006], [Fry 2008], however extending it by utilizing forms, methods and patterns found in living systems such as the organisation of cells, to populations, communities, and ecosystems. Letting the user to navigate through the evolution of engineering information created through product life cycle by dynamic information sources may thrive on this nature mimicking approach. Organic systems are by definition open systems, all of which exhibit responsiveness to stimuli and self-organization and maintenance whilst resource competing [Johnson 2009].

Visualising engineering information that is processed and determined by stakeholders within product's lifecycle could be seen as an analogy case to living systems. Openness can be mapped to creation, change and retirement of information and relations which the visualisation system tries to process in order to display layout structuring. Information structuring could be provided in such way that permits more efficient facts establishing and drawing up of conclusions based on person's pattern identification capability. Interactivity can be seen as reaction to outside stimuli performed by user during the search and navigation through data and information sets. Emergence of self-organization is in this case a bit over exaggerated since the structuring of information will be governed by visualisation system's set of rules and algorithms, but nevertheless it will provide a foundation by which the user can efficiently mount a search process.

A portion of what is tried to be implemented for visualising evolution and traceability of engineering information can be observed in the Visual Thesaurus which is developed by Thinkmap (http://www.visualthesaurus.com/). A dynamic browsing enables user to navigate through thesaurus as walk-through over semantically related terms. After a mouse click to a term represented as graph's node another view unfolds which is centred according to the clicked-on preselected term. Network's structure emerges after applying force field based layout algorithm. Searching for specific term's meaning and relating it visually to similar terms enhances the user's navigation and understanding of the term.

Another example is a Visual Understanding Environment (VUE). It is an open source tool developed at Tufts University [Kumar and Saigal 2005]. VUE provides a visual environment for structuring, presenting, and sharing digital information. After the user specifies the term of interest, VUE uses network or Internet available resources to relate and organize supporting ontology addition to provide further semantic meaning. Both examples aim for visual understanding with providing support to the user to help resolve the limitations of working memory in both capacity and duration of stored information [Sweller and Chandler 1994]. Depending on the information resource location network is being composed thus enabling navigation through content. Both examples exhibit responsiveness to user event based interaction; network is displayed with labels to allow meaning interpretation.

Before proceeding further with the visualisation environment selection an approach originating from Design research community studying collaboration in large design teams [Dong and Moere 2005] is worthwhile mentioning. Most often the engineering visualisation application aim at qualitatively analysing large data sets offering some interactivity, e.g. multi-objective optimisation applications [Parmee and Abraham 2004], but this approach utilises a three dimensional visualisation to study and understand dynamics of large-scale design team collaboration. In their work it was pointed out that conventional project management representations of organisation charts and graphs about deliverables are not suitable to provide understandable foundation to establish how the team is performing on a social level. The visualisation technique was rule based and swarm intelligence driven in order to measure the level of collaboration and to simulate team self-organization as collaboration with shared understanding about problem emerges.

3. Existing visualisation environments

In order to develop a visualisation for TRENIN various visualisation environments and tools have been analysed. The goal of the analysis was to select a well documented open source environment which should possess development features of modern programming environments The environment that is to be selected should provide high quality rendered visualisations as stand alone or as applets, and the framework should be alive with respectable developer community. Leaning to visualisation oriented environments and tools helps to elude demanding low-level programming to produce visualisation environments from the scratch. Summary of visualisation environments and tools comparison is presented in Table 1, which is followed by short overview of visualisation programming environments and tools.

Table 1. Visualisation environments and tools comparison

	Type	Purpose	Open source	Active development	OS	Language	Deployment	GPU support
Protovis	Toolkit	Visualisation toolkit	Yes	No	Requires browser Windows, Linux Mac OS X	JavaScript	Browser applet	As in OS
Gephi	Tool	Dynamic network analysis	Yes	Yes	Windows, Linux Mac OS X	Java	-	As in OS
MSAGL	Tool/ Control library	Complex graph modelling and representation	No	No	Windows	.NET C#	Stand-alone or library import	As in OS
Processing	Environment	Visualisation environment	Yes	Yes	Windows, Linux Mac OS X, Android	Java based	Applet or stand-alone	Direct OpenGL support

Protovis (http://vis.stanford.edu/protovis/) is free and open-source graphical toolkit developed by the Stanford Visualization Group. It allows some control of graphical systems by dealing directly with graphical objects (i.e. shapes, lines), but also enables higher-level constructs such as layouts to simplify recurring programming code lines. Protovis' support ranges from simple views of data such as bars and dots to more complex data structures like graphs and their adjacency matrices. It supports code reuse through inheritance, interaction and animation. Protovis uses JavaScript for web-native visualizations. No plug-ins are required except for a standard web browser. The latter is a major drawback since stand-alone visualisation packages cannot be deployed, thus not fully utilizing hardware acceleration provided by the graphic processing unit.

Gephi (http://gephi.org/) is an open-source and free interactive application for visualization and analysis of networks, complex systems, dynamic and hierarchical graphs. It is a tool which enables users to model, explore and understand massive networks and perform dynamic network analysis. Through interface the user can interact with the graph representation; apply various layouts and metrics. Graph representation is limited to simple directed graphs.

MSAGL (http://research.microsoft.com/en-us/) is an automatic graph layout tool developed by Microsoft Research. It is both an application and control library, thus offering all of the benefices of Microsoft .NET Framework as to integrate MSAGL into any application that requires graph visualisation. MSAGL is able to visualise complex directed graphs, such as those found in business management, manufacturing, and network analysis, as well as phylogenetic trees, which are used in bioinformatics research. MSAGL is not free, and requires a purchased licence.

The Processing (http://processing.org/) [Fry 2008] is an open-source and free development environment, and as said by its founders, it is a visualisation environment for programmers employing organic information visualisation. Initially conceived as an extension to Java, Processing outgrew to include full framework for complex data visualisations, artistic installations and motion graphics. It deploys applets for the web or stand-alone applications on various platforms including the mobile ones. Experimental versions of Processing included extending functionality to other programming languages and frameworks such as JavaScript or Python. A collection of libraries supports more advanced features such as drawing using OpenGL back-end, reading XML files, and exporting in PDF format.

At the end of our analysis the Processing came out as the *most widely* applied visualisation environment in information analysis, industrial design and architecture. Its most significant features are easiness in deployment of applets and stand alone packages for multiple operating systems and platforms, a direct plug-and-play application of OpenGL graphics libraries for enabling hardware acceleration on the GPU and a possibility to add Java compiled extern libraries which is essential when developing larger projects.

4. Engineering information evolution and traceability visualisation results

For the purpose of organic visualisation of the *Traceability Record* development a labelled directed multigraph provides the underlying mathematical model. Multi graph is applied since it supports multiple relations among nodes enabling multiple semantic relationships between the entities of the record. Dynamic information system modelling using the same type of graph is already implemented by W3C for design of resource description framework (RDF) by which online available information are conceptually described and modelled [Klyne and Caroll 2004.].
Traceability reference model [Štorga et al. 2011[a]] is considered in our project as a prototype information model for building semantic *Traceability Record*. A traceability reference model identifies physical and abstract concepts and relations from the product development domain relevant for description of the information object in the context of the product development process. *Traceability Records* should not only help to trace information objects related to the products of development (product information), but should also enable traceability on the activities on related information objects (dependencies) to maintain understanding of the information objects development context (process information). During *Traceability Episode*, entities from the reference model – *Traceability Elements* (TE), are mapped to the information objects managed by external applications (e.g. EDM/PDM/PLM) – *Traceability Objects* (TO) like projects, products, documents, items, users, flow processes, etc.. The *Traceability Events* (TEV) are driving execution mechanisms of the *Traceability Episode* and are initiated by the events on information objects managed by external application (e.g. new, release, approve, update, etc.) or are generated manually by the user during the episode execution (add element, add link, add object, etc.). Thus, each of the graph's nodes may represent *Traceability Record*, *Traceability Element* or *Traceability Object*. All of these can be related using various semantic relations described in reference model – *Traceability Links* (TL). Classes of relations, elements and objects which may appear are prescribed by *Traceability Ontology* that was developed as a part of the TRENIN project.
When put in relation to organic information visualisation technique, the visualisation of engineering information evolution recorded by *Traceability Records* is defined by following features:

- **Structure** - an aggregation of elements to form more complex structures; in TRENIN a directed labelled multi graph is applied for structuring of TR; it is assumed that forming or clustering into structures is dependent on the basis of TRENIN ontology defined prior or during the engineering information evolution.
- **Appearance** - visual expression of internal state; an internal state corresponds to the moment in time in information objects life continuum – colouring and labelling of nodes is used to convey the meaning. In addition various effects like dimming of graph's portions or performing animation are used to emphasise the import parts in structure or event occurrences.
- **Metabolism** - states of engineering information evolution and traceability from beginning to the end of tracing are shown depending on the event occurrence traced by TRENIN system
- **Growth** - an increase in either scale or amount of structure; in TRENIN it is assumed that that during the *Traceability Episode* related engineering information will increase both in number of relations and objects; zooming for close examination over user interface is provided.
- **Homeostasis** - the maintenance of a balanced internal state; labelled multi-digraph's layout is maintained by a "force field strength" which is calculated based on nodal connections within its neighbourhood. Fluid damping relaxation is applied to achieve smoothness in layout formation after stimuli has been applied.
- **Responsiveness** - reaction to stimuli and awareness of the environment; in TRENIN interaction enables user to apply various filters, direct interaction to graph's entities or indirectly via graphical user interface controls.

- **Adaptation** - adjustments to survive in a changing environment; nodes cannot be deleted but can be hidden - layout will always recalculate in respect to pre-existing structure prescribed by prescribed ontology and non-hidden nodes.
- **Movement** - behavioural expression of internal state; in TRENIN instead of just appearing the nodes always move (shift) to recalculated position according to applying damping algorithm; the structure may be shifted in-plane or rotated.
- **Reproduction** - the ability of entities to create others like itself; new data which enters the visualizated TR in time are positioned depending on the prescribed ontology and already existing structure, thus automatically achieving desired clustering's among related elements.

Example of *Traceability Record's* organic visualisation with corresponding GUI is shown in Figure 2:

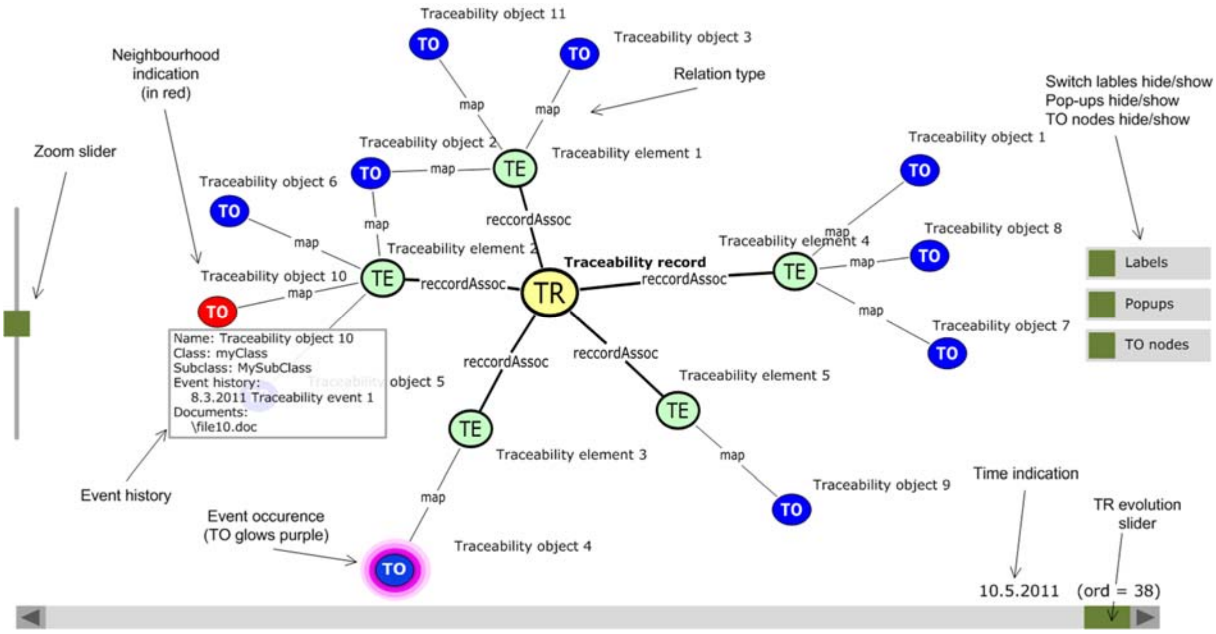

Figure 2. Example of *Traceability Record's* organic visualisation with corresponding GUI

All of the corresponding classes and programming in Java within Processing were developed in-house. Starting from the single node representing a *Traceability Record* user will have to navigate through time line to inspect the evolution of *Traceability Elements* (context) and *Traceability Objects* (content) involving the *Traceability Events*. The input to visualisation is provided by TRENIN Engine application that is part of the TRENIN framework as collection of states (see example in Table 2) that are result of dynamic query on recorded traces. Start and stop points of traceability are relevant for only for *Traceability Record*, and all of the other elements or objects have to became adjunct or created a new and then adjunct to *Traceability Record* within that time frame. Each of the node classes are coloured differently: *Traceability Record* in yellow, *Traceability Elements* in green and *Traceability Object*s in blue and labelled accordingly. The following points in the respect to the visualisation of the TRENIN Visualisation Toolkit are so far achieved:

- **Static (dynamical) query as an input** – initial structure evolves from queries which are established on TRENIN's database in order to prepare all the relevant information related to the individual Traceability Record.
- **Exploration of TR's structure in time** – engineering information grows over time both in number of elements and objects related and number of relationships among them; it is required to show the evolution of engineering information over labelled directed multigraph. By doing so it is possible to establish traceability over the engineering information evolution.
- **Dynamical filtering** – enable hide and show of *Traceability Objects* on foundations of their associations to *Traceability Elements* (and their relations) based on TRENIN ontology.
- **Labels hide/show** – control hide/show of nodes and relations labels.
- **TO/TE/TR attributes inspection** – attributes accessible via pop-up on mouse hover event.

- **Pop-ups hide/show** – to enable hide/show of pop-up window which appear when mouse is over graph's node; pop-up contains attributes of traceability entity (TR, TE or TO).
- **TO hide/show** – enable hide/show of *Traceability Objects* thus letting user to concentrate only at the portion of the TRENIN ontology that has been associated with the *Traceability Record*.
- **TO events history/event occurrence visualised** – occurrences of the events in time are visualised by purple glow around *Traceability Object* on which the event occurred.
- **Applying force vector/fluid damping graph relaxation algorithm** – automatic layout is achieved through relaxation algorithm. Besides the clustering of objects of same type, the self-organization is governed by the relations between *Traceability Object* and Traceability *Elements*, i.e. the class relationships.
- **Zoom/time navigation controls** – enables user to zoom to desired portion of *Traceability Record* in specific time instance.
- **User interaction on mouse hover** – user event management which enables control interaction and pop-up appearance.
- **Semi-transparent interactive pop-ups** – semi-transparent windows enable user to retain the visual context to the data which would be otherwise covered by the appearing pop-up window.
- **Neighbourhood identification** – enables user to see the neighbourhood of a node exemplified and by that to easily recognize the context of selected node.

5. Case study

As a test case study for validation of the proposed visualisation methodology and developed tool, the product development project on new generation of the active head support for car seats from our industrial partner has been selected. Due to the intellectual property protection agreement we are not able to present the test case details but we could provide general information. Ergonomically designed, the car seat head support encourages a relaxed sitting posture and can help with neck pain relief, low back pain, tension and fatigue for driving. The car seat head support offers firm support and promotes a relaxed posture improving driving comfort as well as passenger safety (Figure 3). Development of the head support is strictly constrained by international regulation and standards (UN ECE R-17).

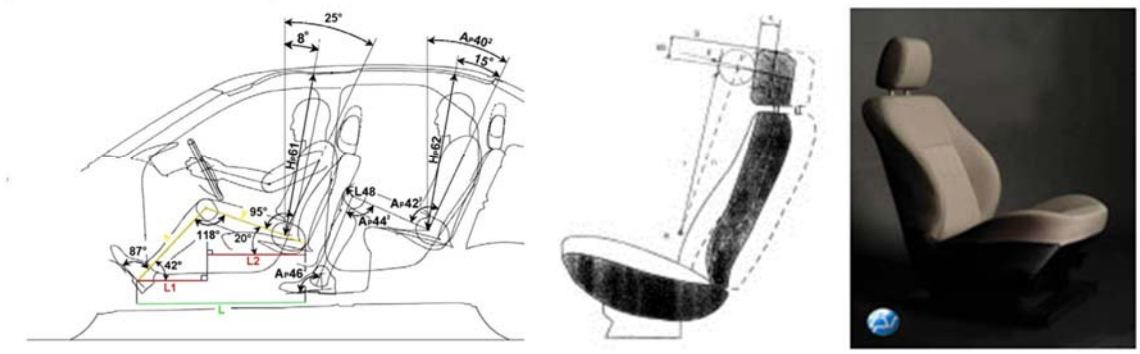

Figure 3. Head support model, testing and application

For the case study purpose, the *Traceability Records* for early design phase have been created using TRENIN framework [Štorga et al. 2011[b]]. The test *Traceability Records* have following characteristics:

- TR based on the combination of the process and product oriented *Traceability Elements*, with a goal to trace project execution and implementation of the safety guidelines and standards in early phases of design.
- Process *related Traceability Elements* have been defined by instantiation of the activities taxonomy (part of the TRENIN ontology extended for particular industrial partner) with the

goal to trace key events during the early phase of development and understand context within the information objects have been evolved.

- Product related *Traceability Elements* have been defined by instantiation of the requirement, function, concept, component and characteristic taxonomies (part of the TRENIN ontology extended for particular industrial partner), in order to explain rationale and background of the design decisions in early phase of design.

The validation objectives of the case study were focused to the demonstration of the TRENIN visualisation toolkit possibilities for:

- Information context evolution visualisation based on the capture of the *Traceability Records* elements, object and relationships dynamic.
- Information objects evolution visualisation based on the capture of on the *Traceability Events* on selected information objects from the external PLM system.

In following table (Table 2) a portion of input data for visualisation in tabular form after querying from TRENIN database is shown. Subclass, responsible person's details, file linking, description, time stamp and ordinal number column are omitted from Table 2.

Table 2. A portion of Traceability Record in tabular form after querying TRENIN database

Type	ID	ID_Object_1	ID_Object_2	Name	Class	Begin	End
TR	TR1			Traceability of the new semi-active car seat head support development		1.9.2009	1.6.2011
TE	TE19			Head support development process			
TR-TE_REL	TR-TE_REL1	TR1	TE19				
TE	TE20			Semi-active head support	Product		
TR-TE_REL	TR-TE_REL2	TR1	TE20				
TE	TE21			Car seat frame	Assembly		
TE-TE_REL	TE-TE_REL19	TE20	TE21	part of			
TE	TE22			Mechanism for positioning of head support	Assembly		
TE-TE_REL	TE-TE_REL20	TE20	TE22	part of			
TE	TE23			Head support frame	Assembly		
TE-TE_REL	TE-TE_REL21	TE20	TE23	part of			
...							

Figure 4 presents screenshot of the engineering information traceability and evolution visualisation based on the input from Table 2, approximately after one third of the traceability episode total time.

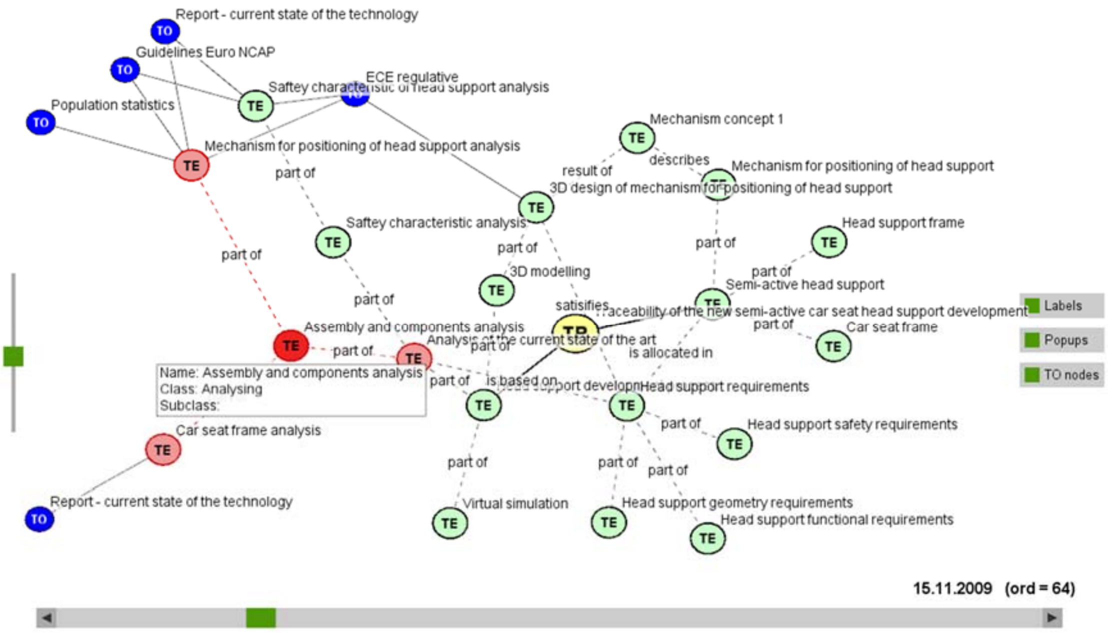

Figure 4. Screenshot of the *Traceability Record* for development of the active head support

6. Discussion

The aim of presented research and development was to support user's cognitive processes through proper visualisation in order to grasp the content of *Traceability Records*. From the literature it is known that it is far easier to have complex concept structure represented externally in a visual display than can be held in visual and verbal working memories [Ware 2004]. Thus, we claim that a visualisation can extend the limits of persons working memory, letting the person to rely on pattern recognition which can be distinguished among entities of *Traceability Records*. From our case study we found that output of the traceability episode for mid-complex electro mechanical product as is semi active head support could be huge amount of recorded data relevant for the engineering information content and context development. We believe that such *Traceability Record* can be better understood and navigated, information is easier to uncover, understand and maintain by visual representation of their interrelations. The experience of the users from our industrial partner has confirmed that it is necessary to visually represent the *Traceability Records* for better understanding of the engineering information development. For correct interpretation of the recorded traces it is essential to have a visible record structure together with the interfaces and procedures for searching and/or navigating among the records.

Also, the users pointed out the shortcomings that helped us to specify features to be included in order to improve features and ergonomic aspects of developed:

- **Directed labelled multigraph** – currently only single graph is supported; displaying multiple relations between nodes requires modelling relations as spline not as simple line.
- **Pan/rotate control** – in plane pan/rotation to change view focus.
- **Ability to zoom on any node** – zoom to any node can be achieved by allowing the shift of view focus via pan functionality.
- **Ability to navigate another TR if related to considered TO/TE** – allows user to dynamically navigate to and to browse another TR if related to considered TO/TE.
- **Ability to rearrange layout without imposed force/damping** –possibility to turn on/off relaxation algorithm to "untangle" some of the resulting structures.
- **Dynamical/user desired font size management** – let the size of labelling font be determined by the zoom.
- **Advanced filtering (including filter over relations)** – to ease the understanding with large datasets.
- **TO colourings/shaping depending on information object type (reports, CAD, emails, etc.)** - to enhance users understanding and pattern recognition.
- **Label font size management** – although labels help to convey the meaning, their size may hinder the process being too small to read or to large thus overlapping with other labels.

Besides TR based visualisation view two more visualisation should be added: a TO based visualisation and TE based visualisation. A query could start with TO showing its relationship over all of existing records and elements. As well the query could start by selection of TE and displaying all of the objects of selected class existing over all of possible records. The same functionality which exists to TR visualisation should apply in these two cases.

7. Conclusions

This paper is presenting an organic visualisation based approach to visualisation of engineering information evolution and traceability during product's lifecycle. It is a part of the TRENIN research project aimed to development of semantic *Traceability Records* for integration of process and product information that is fragmented across different information objects and managed by PDM/PLM environments. To understand the meaning which emerges across engineering information within *Traceability Record* recorded as a complex semantic network, it was soon realized that utilisation would require more than table based database query outputs. The result presented is the TRENIN Visualisation Toolkit which employs organic visualisation technique to visualise engineering information evolution and traceability. Employing living organism's features like self-organization and adaptation for information analyse is a way to present information structured in meaningful and coherent way. Dynamical information provided as input is displayed by TRENIN Visualisation Toolkit as interactive and visually refined. Toolkit is developed in the Processing environment which is allowing reasonable programming effort with straightforward OpenGL graphics library utilization and possibilities for multi-platform deployment. TRENIN visualisation toolkit enabled users to understand engineering information content and context development by providing visible record structure together with the interfaces and procedures for searching and/or navigating among the records. The shortcomings indentified marked the further development as to include various filters, provide pan/rotate controls and to visualise multiple relations between nodes.

References

Bolton, R., Hand, D. J. and Crowder, M., "Significance tests for unsupervised pattern discovery in large continuous multivariate data sets", CSDA 46(1): 57–79, 2004

Card, S.K., Mackinlay, J.D. and Ben, S., "Readings in Information Visualization: Using Vision to Think", San Francisco: Morgan Kauffman., 1999

Dong, A., Moere, A.V., "Visualising Collaboration in Very Large Design Teams", Proceedings of the 15th International Conference on Engineering Design – ICED'09, Melbourne, The Design Society, Australia, 2005

Fry, B.J., "Organic Information Design", Master thesis, Massachusetts Institute of Technology, 2000

Fry, B.J., "Visualizing Data", O'Reilly Media, Inc., 2008

Johnson, N., "Simply Complexity: A Clear Guide to Complexity Theory", Oneworld Publications, 2009

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

Klyne, G., Carroll, J.J.,(Eds.), "Resource Description Framework: Abstract Syntax", W3C, 2004

Kumar, A., Saigal, R., "Visual understanding environment", JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, ACM, 2005

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

Parmee, I.C., Abraham, J.A., "User-centric Evolutionary Design", Proceedings of the 8th International Design Conference – DESIGN 2004, Dubrovnik, Croatia, 2004

Plötzner, R., Lowe, R., (Eds.), "Special Issue: Dynamic visualisations and learning", Learning and Instruction, 14, 235-357, 2004

Štorga, M., Bojčetić, N., Pavković, N., Stanković, T., "Traceability of Engineering Information Development in PLM Framework", International Conference on Product Lifecycle Management, 2011[b]

Štorga, M., Darlington, M., Culley, S. J., Marjanović, D. "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 University, Stanford, USA, 2009

Štorga, M., Marjanović, D., Savšek, T., "Reference model for Traceability Records implementation in en.gineering design environment", Proceedings of the 18th International Conference on Engineering Design ICED11,Copenhagen, Denmark,2011[a]

Sweller, J., Chandler, P., "Why some material is difficult to learn", Cognition and Instruction, 12 (3), 185-233, 1994

Unwin, A., Theus, M. and Hofmann, H., "Graphics of Large Datasets – Visualizing a Million", Springer Science+Business Media, LLC, 2006

Ware, C., "Information Visualization: Perception for Design", 2nd Edition, Morgan Kaufman, 2004

Dr.sc. Tino Stanković, dipl.ing.
University of Zagreb
Faculty of Mechanical Engineering and Naval Architecture
Chair of Design and Product Development
I. Lučića 5, 10000 Zagreb, Croatia
Telephone: (+385 1)6168 369
Telefax: (+385 1)6168 284
Email: tino.stankovic@fsb.hr
URL: http://www.fsb.unizg.hr