INTERNATIONAL DESIGN CONFERENCE - DESIGN 2004
Dubrovnik, May 18 - 21, 2004.

TRACEABILITY IN PRODUCT DEVELOPMENT

Mario Štorga

Traceability, product development, ontology

1. Introduction

During the last decade products have become more and more complex. Their expected usage period has grown, whereas the expected time-to-market for introducing changes becomes shorter and shorter. In order to master these challenges manufacturing companies have provided approaches for reusability, adaptability, and variety of product and partial design solutions. Fundamental problems of such approaches are lack of design solutions understanding and the danger of mistakes during solution adaptation and integration. The problems are mostly caused by insufficient design documentation, and inadequate support to the tracing of design evolution. In practice engineers mainly doesn't record justification for design decisions and the reasoning lying in the background of those decisions.

Why is achievement of product development (PD) context [Štorga 2004] traceability in modern highly-automated, information-overbookedp product development paradigm, still so difficult? We contend that the reason has as much to do with processes and human factors as with issues of heterogeneous tools and distributed teams. Things are traceable, if they leave traces (*Ger.* Spur). In a modern product development paradigm people exchange PD context across corporate boundaries, and reuse existing knowledge in extended meaning. Because of lack of the formal representations of the complex design objects, these exchanges still partly occur informally (face-to-face across a table, by phone, by paper). As a consequence, retrieving of the design objects' information (e.g. with respect to format, type, and contents) as well as correct interpretation (due to the specific domain), hinder the product innovation and produce unnecessary development iterations.

During product development designers need traceability carried by traces of the design routes, because they want to reuse existing design knowledge along meaning, reasons, arguments, documentations (proofs, specifications), choices, critique, consequences etc. One of the major problems in modelling of design knowledge on design routes is in finding an appropriate set of formalized concepts that the knowledge should refer to [Štorga 2004], or in more fashionable term, ontology. These concepts and relations between them should be general enough for fully describing the different design data, information and knowledge in different design domains, but specific enough to do justice to the particular nature of the task we are discussing about: traceability in product development.

This paper attempts to offer some answers on open questions in tree steps. Firstly, we have studied the nature of traceability in product development (Section 2) what was aimed at clearly identifying of basic objectives and implementation difficulties of traceability in product development. Secondly, we proposed meta model of PD context for achieving traceability based on classification of the main traceable items and links according to literature overview (Section 3). Finally, we discussed product development ontology as a framework for traceability implementation (Section 4). We conclude with presentation of the main results and of the directions for the future research (Section 5).

2. What is traceability?

2.1 Definitions and dimensions

Traceability can be defined as a quality factor of designing – a property that product development environment should possess and include as a non-functional feature. More comprehensive picture of traceability can be given through considering of the main elements and conditions for traceability application in product development environment (Figure 1).

Firstly, traceability can give essential assistance in management of the requirements, that are results of the needs or predicted future meetings between design and the different life phase systems (in production phase, using phase, servicing phase, etc.). Traces of requirements evolution and verification procedures should help designer in ensuring of the requirements fulfilment and keeping track on the development project status.

Secondly, we can look at the definition by Hamilton and Beeby [1991] who view traceability in product development as the ability to "discover the design history of every feature of a product". Management of the design history should allow a following of the evolution of a design items, in both a forward and backward direction, i.e. from its origins, through its development and specification, to its deployment and realization, and through periods of on-going refinement and iteration in any of these phases. Also, tracing of the design history should improve understanding of the design routes by linking designed items to justifications, important decisions and 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.

Finally, we can say that traceability is a property of a product development environment with a goal to ensure that PD context (*data, information and engineering knowledge that evolves throughout the product development*) is clearly linked to its sources in design representation created as the result of the product development life-cycle. PD context documentation is seen as the main objects from/to which traceability should be performed during the product development. They indeed have a central role in development: they describe and document constitution and behaviour of design; they drive the design and are the object of verification and validation procedures. Aiming of a traceable designing is to help in ensuring that PD context documentation and links set between them are complete, correct, consistent and error free.

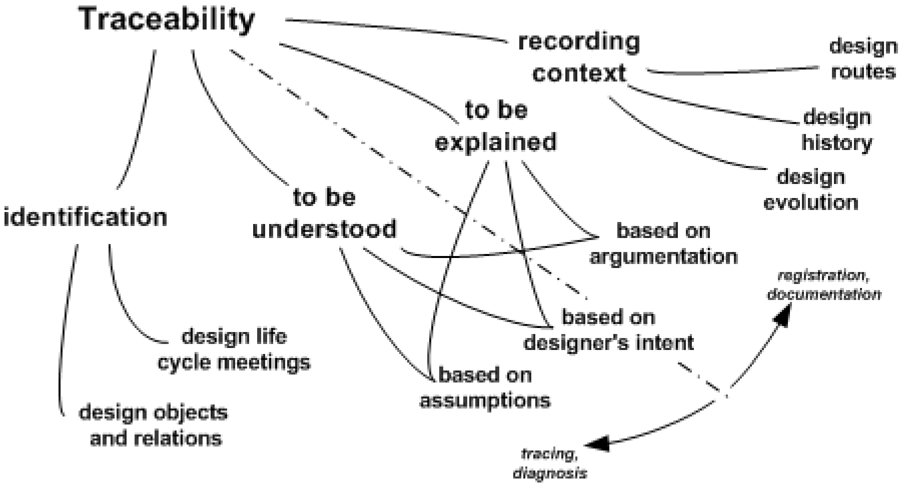

Figure 1. Main elements and conditions for traceability

Many different actors in product life-cycle are involved in the product development process. As a consequence of these different uses and perspectives on traceability, there are wide variations in the format and content of traceability forms across different product development efforts. The current literature and the standards do not provide detailed guidelines of what types of PD context must be captured and used in what meaning. In our observation on discovering the main dimensions of traceability contents, we can start from assumption that the traceability is always related to a specific design episode. Such reasoning comes from re-consideration of the design decision-making framework proposed by Hansen and Andreasen [2000]. Each design episode has the specific traceability scenario that can be described by answering the basic questions:

- **What** are the traceable items —objectives, requirements, design representation objects, tests, etc.— that are managed during design episode, what are their characteristics and what are the links between them?
- **Who** are the actors that play different roles in the creation, maintenance and use of those items and links?
- **Where** are the items represented: in physical media such as documents, drawings, spreadsheets or other files, or in less tangible things such as telephone calls, discussions, emails, meetings or people's brains?
- **How** are they represented (by more or less formal means)?
- **Why** were they created, modified or evolved?
- **When** were they created, modified or evolved?

2.2 Main traceability difficulties in product development

Based on literature review, we have identified several difficulties for achieving full traceability in product development:

1. The situated nature of traceability creates variety of traceability needs. The consequence of fact that traceability is ruled by design phenomenon is that traceability content and needs varies greatly from one organization to another, from one project to another and even from one user to another. Traces providers – users that perform traces production (mainly the members of the development team), and other users - those that perform traces extraction (managers, other development team members, test and maintenance team members) - have different goals and priorities in applying traceability on designed items and design routes [Ramesh 2001].

2. Knowledge base for a design to be fully traceable is big and complex. Because the traceability contents can vary widely and because the multirelations and dependencies between designed items, applying traceability in product development results in big and complex knowledge base. Creation and maintenance of such base require a lot of effort. Besides, traditional techniques for locating and maintaining the sources of design items are not sufficient to achieve full traceability, because sometimes company rules prohibit the access and knowledge of the original sources. The main consequence is a slow realization of later changes, because the identification of those to involve and inform is erroneous and time-consuming [Riddle 2002].

3. Traceability needs support from both formal and informal sources. Because the product development is primarily a collaborative process, most development efforts lack well-defined formal models for describing process and design. Informal sources are easy to capture but the classification, indexing, retrieval and use of such sources can be impractical and it is not liable for automated reasoning. Reducing inherently informal to a formal representation may facilitate their management but much of meaning embedded in such sources can be lost. That is why decision whether to formalize or not rests on cost-benefit analyses, stability of the content, and the question whether some portion of design knowledge can reasonably be formalized at all.

4. Traceability overlaps with existing heterogeneous notations and tools. There are no existing tools that support achievement of the full traceability in product development. The existing tools besides their main functionality provide possibilities for the partial traces production, extraction and/or verification. For example, PDM tools allow navigating some historical routes of a product development by recording the evolution of different documents through main revisions and versions. In the same time the feature based CAD tools provide possibility for tracing history of building geometrical feature tree. But, the heterogeneity of existing tools serves as a reason for difficulties in full traceability implementation. "In engineering, each discipline has its own languages, methods and tools. This results in a lack of ability to trace across disciplines" [Palmer 2000].

2.3 Traceability achievement

Research and support of traceability in product development is immature. There is a lack of common understanding of what traceability in product development is, what its objectives are and what problems it should solve. Despite the importance of this topic, there is surprisingly little written about it, the standards and literature provide little guidance and do not provide detailed guidelines on what design items must be captured and in what meaning [Ramesh 2001]. The absence of automatic techniques to assist in the identification of design routes and the lack of effective support provided by tools, imply that attempts for producing traces as well as for extracting them can take too much time.

Considering how to achieve traceability in product development, we can emphasize the existence of the three main stages:

1. *Identification and planning.* This stage is characterized by definition of traceable items accordingly to design episode (Section 2) where the designer decided to make things traceable. In this stage, the main task is to define what (kind of) objects should be traced and what (kind of) links are needed between those objects.
2. *Recording and documentation.* The main task of this stage is in creation of traces that are result of product development activities, designers' actions, decisions, reasoning, events, etc. Only explicitly defined design items and routes should be recorded and documented for further use. It is performed during all phases of product development.
3. *Utilization.* The main task of this stage is extraction of design items by following, querying, and navigating them through recorded and documented traces. It is done by simulating design episode in another situation: for performing changes on existing solutions, reusing of the existing solutions in new projects, configuration of the new variant of the product, and educational process for inexperienced designers.

The analysis of the considerations presented in previous sections, lead authors to conclusion that well defined syntax and semantics of design items and design routes representation, can be a first step in successfully traceability achievement. We also conclude that the more semantics are defined for routes and design items, the more "intelligent" treatments can be performed on their traces. Thus, both syntactic and semantic information are needed to successfully implement tracing, because it is not enough to know only the form, it is also necessary to know the meaning of traces. As the result of our research direction, meta model of PD context for achieving traceability is proposed and explained in the following section.

3. A meta model of PD context for achieving traceability

The meta model of PD context describes all construct needed to generate traceability models as well as their semantics. It is builded following dimensions necessary for achieving traceability in design episode (Section 2). In definition of building blocks, we assumed that proposed meta model will be implemented in framework based on product development ontology [Storga 2004]. The meta model consists of traceability items and links (Figure 2), and each of them can be specialized and instated to create specific implementation models.

3.1 Traceability items

On the basis of the collected findings (Ramesh 2001, Delannay 2003, Hobbs 2003, Mortensen 1999) and discussions in research group, the main items of the meta model of PD context for achieving traceability in product development, were determined (Figure 2):

- *Design objects* are generalization of inputs and outputs of product development process. Following the research results on developing the Genetic Design Model System (Mortensen, 1999.) (because it's more comprehensiveness comparing with other design model systems that can be found in literature), the *design object* can be various types: *requirements, technologies, processes, organs, functions, parts, design characteristics, design properties*. These items represent major conceptual elements among which traceability should be maintained during the product development.
- *Subjects* are generalization of different actors and their roles involved in the product development. The *subjects* can be project managers, designers, analysts, customers, suppliers,

testers etc. *subjects* act in different roles in creating, using and maintaining various *design objects*. Roles are characterised by *subject* knowledge, competencies, rights and obligations.

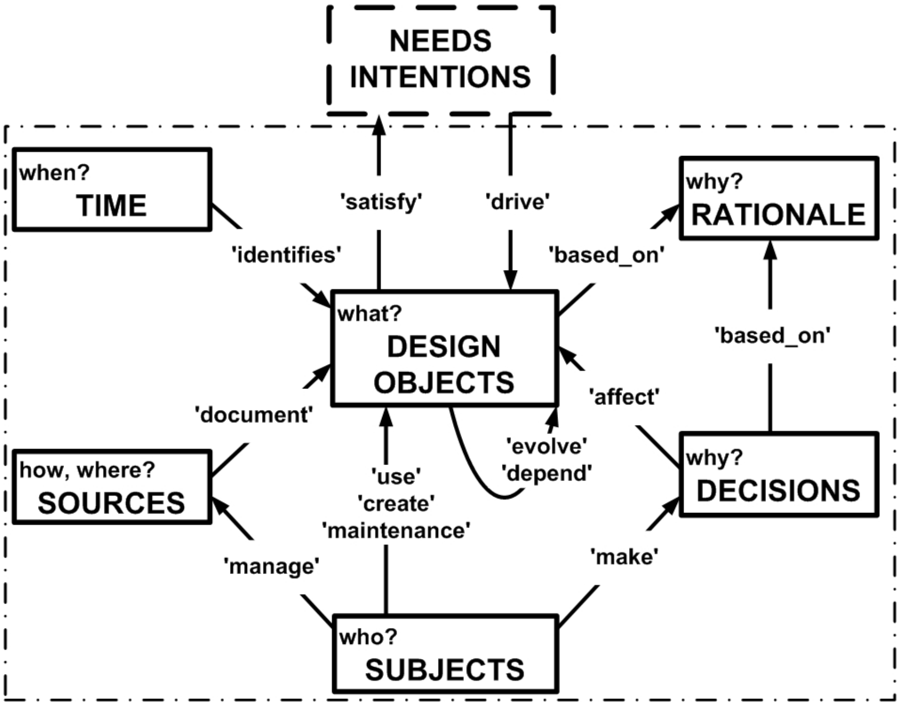

Figure 2. The meta model of PD context for achieving traceability

- *Sources* documents the *design objects*, and may be different physical media, computer readable documents, references to the people or undocumented policies and procedures. There exists a remarkable diversity in formats and levels of formalization between different source types.
- *Time* identifies the *design objects* by two dimensions:
  - *relative time* dimension described by proceed-follows and stimuli-response relations between different *design objects* in order to show how specific traceability items are ordering with respect to one another, and how they following each other during the execution time (meetings, processes, organs, functions);
  - *absolute time* dimension that anchoring traceability items in time line by different attributes as state, stage, status, version, revision, timestamp, in order to show and capture the history of development process, *decision* making process, or evolution of *design objects*.
- *Decisions and rationale* are representation of the design rationale behind the creation, changing, and evolution of the various *design objects*. The information about how decisions are made by *subjects* must be maintained during the product lifecycle to ensure that *needs* are understood and satisfied. *Decisions* based on *rationale* affect *design objects* by evaluating and selecting its alternatives.

3.2 Traceability links

Formally speaking, a proposed meta model for achieving traceability can be defined as a semantic network in which nodes represent traceability items, among which traceability is established through links of different types. In the literature a large number of link type have been proposed, many of which are also important for traceability. As a starting point for classifying the links in our observations, we adopt a simple system of traceability links accordingly to (Ramesh 2001):

1. *Satisfaction links.* The main purpose is to ensuring that the *needs* are satisfied by *design objects*. The uses of this links are to track the *design objects* to the defined constraints or goals, and ensuring that all *needs* are satisfied by specific product features.
2. *Dependency links.* The main purpose is to manage different kind of dependencies among different kind of *design objects*, in the same level of abstraction (domain), or between *design objects* in different domains. The use of this links is to track composition and hierarchies of *design objects*.
3. *Evolution links.* The main purpose is to capture evolution of the *design objects* from existing to the new or modified. The uses if this links is to track where do the various *design objects* come from, where they are anchoring in a *time*, how looks like modification and refinement history of various *design objects*.
4. *Representation links.* The main purpose is to manage different representation *sources* of *design object*. The use of this link is to track hierarchy and composition of *sources*, and manage responsibility of different *subjects* for specific *source*.
5. *Rationale links.* The main purpose is to represent the *rationale* behind the *design objects* or document the reasons for *decisions* on evolutionary steps. The uses of this links is identification of the reasons behind the creation or modification *design objects*, provided by different *subjects*, capturing important *decisions*, providing transparency into decision process including overview of *design object* evolution alternatives.

4. Framework for traceability implementation

4.1 Product development ontology

Research presented in this paper is a part of the project aimed to the building of generic informational models as a framework for formal representation of the PD context [Storga 2004]. After the first phase of the research, main conclusion is that building of the *product development ontology* (PDO) [Storga 2004] can be the step to achieving such framework for solving the communication and integration problems for dynamic and distributed product development environments.

We propose building the PDO because, firstly, it allows a definition of basic architecture for the PD context management and provides inference engine for different domains in the product development. Secondly, ontology independency helps to resolve the complex principles behind design as a phenomenon, based on the need for formalization of design theories for product development domain [Mortensen 1999]. The ontology, instantiated for a particular PD domain has to be sharable for all subjects (both the humans and computer applications) involved in a development process.

The motivation for building PD ontology can be summarised as the need for:

- universal semantic foundation for the terminology employed between interacting subjects;
- an integrating model for the different views on and aspects of a design;
- an unambiguous definition of the meaning of all concepts employed.

The (explicit) ontology may take a variety of forms, but it will necessarily include a vocabulary of terms and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related and constrains on the possible interpretations of concepts. With previously findings, space of the intended uses for *product development ontology* can be divided as follows:

- Communication medium for achieving interoperability during product development [Storga 2004], between:
  - different people, including users and developers, across different organizational units;
  - people and implemented heterogeneous computational systems;
  - heterogeneous implemented computational systems.
- Building infrastructure of the framework for applying traceability on:
  - acquisition, representation, and manipulation of product development context;
  - structuring, evolution and organising product development context;
  - explanation of the design rationale.

Product development ontology that we are building is centred on two conceptual viewpoints: product life cycle meetings and product representation on different levels of abstraction (product domains) [Mortensen 1999]. The ontology formalization is not finished, and it is expected that will be presented in the future. The meta model presented earlier in this paper (Section 3) will be applied in existing ontology structure by extending it with definition of the new concepts and new attributes for existing concepts. The relations between different ontological concepts will be classified and described accordingly to the five proposed traceability link types (Section 3.2), in order to support traceability implementation.

4.2 Proposed implementation framework

Following the literature (van Elst 2000), the possible traceability implementation framework is depicted on the Figure 3. This approach models and executes development process and tasks on the *application level*. When a user in product development process recognizes need for the *PD context* within the actual flow of work, a semantic query to the *description level* must be derived. This query is instantiated and constrained as specifically as possible on the basis of the actual user needs. In the opposite way, the user can also store new *PD context* created within a given working situation in a contextually enriched form. One of many possibilities for realizing the *application level* can be conventional business-process models and workflow- management systems.

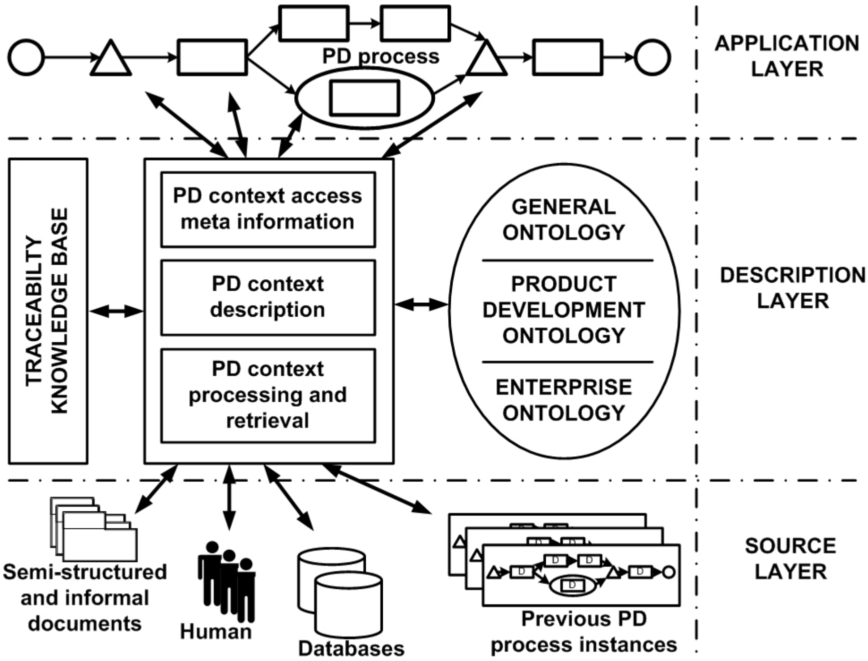
APPLICATION LAYER
PD process
TRACEABILTY KNOWLEDGE BASE
PD context access meta information
PD context description
PD context processing and retrieval
GENERAL ONTOLOGY
PRODUCT DEVELOPMENT ONTOLOGY
ENTERPRISE ONTOLOGY
DESCRIPTION LAYER
Semi-structured and informal documents
Human
Databases
Previous PD process instances
SOURCE LAYER

Figure 3. Proposed implementation framework

The *description level* serve for the mapping from the application-specific information needs to heterogeneous *source level* via a uniform access and utilization method on the basis of a logic-based *product development ontology*. Actual work context is mapped onto expressions in the *traceability knowledge base* resulting with the appropriate assertions and queries. The vocabulary for the *description level* comes from the *product development ontology*, extended by attributes for applying traceability, accordingly to the proposed meta model (Section 3) – such as timelines, sources, evolution links, specific dependencies between design objects etc. Because *description level* relies substantially on existing *PD context*, the *source level* is characterized by a variety of sources, heterogeneous with respect to several dimensions concerning form and content properties. This level comprises manifold information and knowledge sources, ranging from machine-readable formal representations to human-readable informal representations.

The proposed implementation framework built around *product development ontology* provide a features that can help in solving of the existing difficulties for achieving traceability (section 2): (i) ontology provide formal syntax and semantic definitions; (ii) different ontology abstraction levels can be used for avoiding the situated nature of traceability; (iii) semantically rules and deduction mechanisms can resolve complexity of traces among design routes; (iv) semantic interoperability based on ontology can be a communication medium between heterogeneous design object sources.

5. Conclusion and further work

The proposed approach to the achievement of the traceability in product development is at a conceptual stage. There are still many open questions and some unclear points to be clarified in a future. The aim of the authors was to make a first step into the complex problem area with the new ideas, for getting response and advises for further work. It is believed that future research in this field will lead to interesting clarifications in more different areas: design rationale patterns for representing common elements of design reasoning structures in s form minable to trace and reuse, distributed traceability mechanisms focused on making context among heterogeneous systems reusable via semantic mapping and context management, and active traceability knowledge bases with mechanisms that should note engineers if their area of interest is affected by any new developments.

Acknowledgement

This research is part of funded project "Models and methods of improving the computer aided product development" supported by the Ministry of Science and Technology of the Republic of Croatia. The presented paper is result of the research performed in co-operation with the scientific staff of Section of Engineering Design and Product Development, Technical University of Denmark. Author thanks prof. Mogens Myrup Andreasen for his effort and discussions.

References

Delannay, G., "A generic traceability tool", http://www.info.fundp.ac.be/~pth/fundpdocs/gde.pdf, 2003.

Hamilton, V.L., Beeby, M.L., "Issues of traceability in integrating tools", Processdings Colloquium IEE Professional Group C1, London, 1991.

Hansen, C.T., Andreasen, M.M., "Basic thinking patterns of decision-making in engineering design", International Workshop on Multi-criteria Evaluation, MCE 2000, Neukirchen, pp 1-8, 2000.

Hobbs, J., Pustrjovsky, J., "Annotating and reasoning about time and events", http://www.cs.rochester.edu/~ferguson/daml/hobbs-pustejovsky.pdf, 2003.

Mortensen, N.H.: "Design modelling in a Designer's Workbench – Contribution to a Design Language"; PhD thesis; IKS 00.02.A; DTU; 1999.

Palmer, J.D., "Traceability", Software Requierements engineering, 2nd Edition, IEEE computer Society, pp.412-422, 2000.

Ramesh, B., Jarke, M., "Toward reference models for requierements trceability", Software engineering, 27(1), pp. 58-93, 2001.

Riddle, S., Arkney, P., Mason, P., "Position paper: Enabling Traceability", First International Workshop on Traceability in Emerging Forms of Software Engineering, Edinbourgh, 2002.

Štorga, M., Marjanović, D., Bojčetić, N., "Considerations on IT systems inteoperability in product development", Proceeding of the TMCE 2004, Millpress, Rotterdam, ISBN, 2004.

Van Elst, L., Abecker, A., "Ontologies for information management: balancing formality, stability, and sharing scope", Expert Systems with Applications 23, pp. 357-366, 2002.

M.Sc. Mario Štorga, Dipl. Ing. M.E.
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Chair of Design
Ivana Lučića 5, 10000 Zagreb, Croatia
Tel. +385 1 6168 119, Fax. +385 1 6156 940
mario.storga@fsb.hr