DESIGN PROJECTS' REPOSITORY – A TOOL FOR DESIGN PROCESS GUIDING, KNOWLEDGE CAPTURING AND REUSING

Neven Pavković
Chair for Product Development and Design
Faculty of mechanical engineering & naval architecture, Zagreb
Croatia
neven.pavkovic@fsb.hr

Igor Klarić
KLEX d.o.o., Zagreb
Croatia
igor.klaric@klex.hr

Mario Štorga
Chair for Product Development and Design
Faculty of mechanical engineering & naval architecture, Zagreb
Croatia
mario.storga@fsb.hr

ABSTRACT

Paper presents results of the first phase in development of computer environment for achieving traceability in product development process. Following design process support functions are included in prototype:

- guiding the inexperienced designers;
- organizing and cross-linking the digital documentation generated in the design process;
- evaluating and managing the ongoing projects;
- building a knowledge base about finished design projects through capturing significant data and rationale;
- querying the rationale and documents from previous projects.

The main structure for representing the design process flow and for indexing the design knowledge is the design plan, which comprises a state – action model of the design process. Based on design plan usage, the authors propose a model of knowledge generation, capture and reuse. Two important issues of design repository usage in practice are briefly discussed – the guidance of novice designers and the balance between potential benefits of proposed system and the extra time required from designers to use it. Those are key factors that affect many issues in preparing the terminology, the usage procedures and the interfaces.

KEYWORDS

Traceability, design rationale, knowledge capture and reuse, data organisation

1. INTRODUCTION

In current design support systems, knowledge-level design semantics, such as design intent and design rationale, function and behaviour, etc. are represented in an ad hoc manner (Regli et al., 2004). Most often this information is found in associated design files, annotations, and communications that lack any formal structure. In practice engineers mainly doesn’t record justification for design decisions and the reasoning lying in the background of those decisions.

The design process is here viewed as a sequence of transitions from an initial state of data, constraints and goals to a final state, the description of the artefact being designed. These transitions are allocated to individual participants or teams, and individual or integrated computational tools or models. A design process may no longer be viewed as a static institutionalised structure, but rather as a dynamic network that is constructed in real-time as design goals, needs, priorities, and resources evolve (Wallace et al., 1999). In the framework of an integrated CAD environment, the designer should work with an open toolbox, enabling him to create his own classes and partial models of the design process according to his current needs.

The goal of presented research is to develop a knowledge oriented framework that enables to model the design process flow (and to integrate used software tools) independently of design phase and class of design task.
1.1. Related research projects

The issue of creating and capturing reusable design knowledge in engineering design has been addressed by many researchers from various perspectives. Research efforts span the areas of design and behavioural model repositories, analysis templates, and product lifecycle management consistently promoting knowledge and information capture and reuse over the life of the product (Mocko et al., 2004). In most cases designers wish to reuse captured design knowledge to adapt past solutions and apply these to current problems, and novice designers may wish to understand lessons from previous experiences (Ahmed S., 2005).

Similar treatment of formalism in representation of the design process can be found in the work of Gorti et al. (1998). Their model is implemented as a layered scheme that incorporates both an evolving artefact and its associated design process. They define the design process with five primitive objects: goal, plan, specification, decision and context. Liang and O'Grady (1998) have developed a concept of a design object that represents physical entities such as parts or components as well as non-physical entities, such as design history or vendor information. They build the design process model formalism on a definition of a design model, design objects and design methods.

Proposed model of design process stage is in one part similar to "design workspaces" proposed in Grabowski et al. (1995 and 1996). The "design workspace" is focused primarily on geometric modeling. The model proposed in this work is primarily oriented to improve the design process organization and to integrate the data flow through various software tools used in the process. In other words, this research work is focused primarily to represent the design process in the context of information processing.

McMahon et al. (2005) analyse many issues of capturing the outcomes of the design process as a part of information management for through life product support.

Very interesting considerations about planning the design process may be found in Giaopulis et al. (1995). We must ask ourselves to what extent a design process could be planned? Real design process environments generate dynamic situations - they can change while the plan is being executed in a way that makes the plan invalid (Pollack M.E., 1992). As if this is not enough, real environments may also change in a way that do not invalidate a current plan, but instead, offer new possibilities for action.

2. TRACEABILITY IN PRODUCT DEVELOPMENT PROCESS

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.

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. 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.

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 product development 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?

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 development process across disciplines” (Palmer 2000).

### 2.1. Traceability achievement

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

**Identification and planning:** this stage is characterized by definition of traceable items accordingly to design episode where the designer decided to make things traceable. In this stage, the main task is to define what (kind of) objects should be traced and what (kind of) links are needed between those objects.

**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.

**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

---

**Figure 1** The meta model of product development context for achieving traceability
solutions, reusing of the existing solutions in new projects, configuration of the new variant of the product, and educational process for inexperienced designers.

A meta-model of product development context for achieving traceability

The meta-model of product development context (Štorga, 2004) describes all constructs needed to generate traceability models as well as their semantics. The meta-model consists of traceability items and links (Figure 1), and each of them can be specialized and instantiated to create specific implementation models.

Design objects are generalization of inputs and outputs of different phases/stages of product development process: requirements, working principles, activities, 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 subjects’ knowledge, competencies, rights and obligations.

Sources documents the design objects, and may be different physical media, computer readable documents, references to the people or undocumented policies and procedures. 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. DESIGN REPOSITORY – A PROTOTYPE ENVIRONMENT FOR TRACEABILITY IMPLEMENTATION

This chapter presents results of first phase of development a prototype implementation environment for achieving traceability in product development process. The developed prototype still does not include all elements of the previously described meta-model. Following design process support functions are included in this phase:

- guiding the inexperienced designers through design process scenarios, specific for the particular design environment
- providing the structure for organizing and cross-linking the digital documentation generated in the design process
- providing the data and interfaces for evaluating and managing the ongoing projects
- building a knowledge base about finished design projects through capturing significant data and rationale for steps of each individual task and project
- querying the rationales and documents from previous projects

Relational database environment is chosen as the software development platform for several reasons:

- Relational databases offer powerful search tools and means for management of huge data collections.
- Software market offers a wide price range of relational database environments, mainly varying in network and security performances, while there is no significant difference in modelling capabilities of complex data structures.
- The prototype system is targeted for testing and usage in small and medium companies, thus for the purpose of initial development, a low budget and not too complex software development environment would be sufficient.
- After the implementation model will be fully developed and its usability proven, it could easily migrate to more powerful (and expensive) systems.

3.1. The representation of design process flow

Typical product development is realised in small steps by redesigning the existing products, rather than by designing entirely new ones. The proposed repository model does not make difference between the classes of the design task in the context of the design novelty or the design process phase. The main structure for representing the design process flow and for indexing the design knowledge is the design plan, which comprises a state – action model of the design process (Pavkovic, Marjanovic, 2001). In this research phase the focus has been put on support of adaptive and variant designs. We have tried to integrate design information management tools with
The design process planning and workflow model. It is defined as a model of design process decomposition and the execution sequence. The purpose of the model is to organize and to try to predict the information flow as well as the sequence of execution of process steps controlled by conditions and constraints. The design plan is represented graphically as a collection of nodes and their connections in a directed graph, recorded in adjacency and incidence matrices. The connections between nodes represent the information flow and/or the sequences of node execution. The nodes of a design plan represent design process steps. To "execute" a node means to perform the associated operations. In other words, the main elements of the design plan are steps and actions – each step may comprise several actions. The main features of the proposed model are as follows:

- design plan contains a set of predefined steps and actions for the particular design project (task); new steps and actions could be added during the design process
- for the management purposes steps are linked with project workflow and timeline data
- steps and actions contain elements for recording rationale and for project state evaluation
- each design plan, step and action include a set of links to relevant documentation
- the repository contains a catalogue of all kinds of documents being used in the design process

The term "document" assumes here any kind of digital file (CAD part or drawing, spreadsheet, text, picture, etc.).

The connections in directed graph (Figure 2) are classified as:

- "normal" execution sequence (points from current node to next node)
- "recurrent" (repeated) execution sequence (points from current node to one of previously executed nodes) – this is a kind of prototype model of iteration in design process
- "information dependency" connection – indicates that data sets (of objects that are referenced from connected nodes) are directly or indirectly dependent

The design plan node models one step of the design process, including: checking of preconditions, list of actions, checking the "postconditions" and deciding about the next step.

### 3.2. The process of knowledge generation, capture and reuse

#### Identification and planning

According to Ahmed S. (2005) novice designers require support in identifying what they need to know. One of the primary features of the proposed design repository is to enable the novice designer to access the organized recordings of knowledge and experience of senior designers. When starting a new design project, the designer opens the particular design plan skeleton containing a set of predefined steps, actions and their sequence. A design plan skeleton is the structure that should represent, collect and organize every type of knowledge and data that is or should be known prior to start of a new design project. The term "skeleton" is used to emphasize that this is an initial structure for building the complete recording and documentation of finalized design project. Relying on our own previous research work (Pavkovic, Marjanovic, 2001) we assume that it is possible to create skeletons of design plans for adaptive and variant designs. The most experienced designers should create skeletons of design plans for particular classes of design projects in particular design office. This way experienced designers should generate the knowledge about how some particular project should be solved by their own experience. As previously stated 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.

Figure 3 shows schematically the process of knowledge flow from generation to storage.
Recording and documentation

Through the timeline of design project realisation, the initial design plan skeleton is being updated, upgraded and "filled" with created traces that are result of product development activities, designers’ actions, decisions, reasoning, events, etc. At the end of the design project skeleton is transformed to completed design plan with explicit design items and routes recorded and documented. Such design plan is a complete description of a finished design project. Finished design plans are being stored in archive.

Knowledge reuse

By choosing the appropriate design plan skeleton, the designer accesses and reuses expert knowledge and experience. The other way of reusing knowledge is by extraction of design items from archive of finished projects. Extraction could be done by following, querying, and navigating through recorded and documented traces as also by querying and accessing linked documentation in digital (or some other form). Predefined and organized reuse of stored knowledge in a proposed system becomes an integral part of the design project solving process. Predefined and organized reuse is achieved through special items in design plans that tells the designer what is (or should be) available for searching and where this knowledge should be found. Some simple examples of these situations could be: changes on existing solutions, reusing of the existing solutions in new projects, viewing problems and solutions on previous similar projects, analysing testing reports, etc.

On the basis of all previous considerations the "design projects' repository" is composed of following components (figure 3):

- A set of design plan skeletons
- A set of tools and procedures for working on active project by using the copy of chosen design plan skeleton
- The archive of finished design process
- Methods and tools for indexing, searching and retrieving recorded knowledge
- All the documentation linked from design plan items, - it should be considered as "external

Figure 3 The flow of stored and newly generated knowledge in the process of the proposed system usage
Guidance of novice designers

The guiding function of the proposed system is accomplished through following features:

- design plan gives a novice designer a proposal (or a kind of prediction) of the execution sequence of operators in the design process domain, or in other words a guidance (a path) of the progression in the particular design process,
- references to relevant documentation for each design process step and action provide the mechanisms for easier organizing in the process of gathering necessary data and information,
- querying, searching and retrieving the archive of previous projects helps the novice designers to gain the necessary experience and knowledge much faster than by other ways of informal communication with experienced designers – but that doesn’t mean that we underestimate the value of “face to face” transfer of knowledge.

3.3. Entity-relationship model of the proposed system

This chapter gives a brief overview of main entities and their relationships in the prototype database that is still under development.

Once outlined and established, the computer based design process model will be the subject of permanent improving and maintenance processes caused by new design science cognitions or changes in the environment where the design process proceeds.

Therefore, it is proposed to outline the global design process model structure as an "open toolbox". Such an approach should enable the designer to create his own classes and partial models of the design process according to his current needs. It can't be expected that it is possible to build a model general enough to enclose all phenomenon variations in different real environments. Keeping this in mind, the presented research aims to develop a framework that could at least enable the use of different partial models (from design process domain) in an integrated manner.

A real system (real world domain) whose modeling is being considered here is a teamwork, computer supported design process which uses computer network (intrantet) for team member collaboration and data share. Is it supposed that such environment intensively uses CAD software, databases, PDM and other kinds of software tools.

In outlining the design process model the focus will be in efforts to adopt the model as much as possible to purposes of computer application. In such approach the authors would not try to build neither primarily prescriptive or descriptive model, but some kind of a hybrid. The design process will be treated as information generation and transformation, estimating that the majority of the information is computer stored. In other words, the designing is treated as a process for which a computer support should be modeled by mapping a common working process (from the real world domain) to the domain of object-oriented software system.

Based on the presented approach, the entities of conceptual domain will be proposed. In the process of outlining the model structure, the efforts will be focused to make the model general enough for wide spectrum of application. In the same time the model should contain a rich set of specialization possibilities, to be able to efficiently adapt to specific characteristics of particular application environment.

Let us compare the conceptual modeling (outlining) of common business processes and design process. The majority of business processes are relatively simpler in structure than the design process - therefore it is easier to extract their conceptual model from the real world domain. Most of the transactions and operations in business processes are well defined and determined. In the contrary, the big part of transactions in the design process are very difficult to predict, and impossible to prescribe. The main difficulty of this research, particularly in developing the entities of the proposed model, was (and still is) the lack of the general consensus in design terminology, taxonomy, and typology, as emphasized in Andreasen et al. (2000 and 2002). Design science misses the “CAD theory” as pointed out by Akman, et al. (1990).

We have chosen the approach to build and refine the model "step by step", doing all the possible testing before further upgrading. There are many issues of practical usage that affects the model and generate the need for new entities and attributes. Those needs could be recognized only in stepwise attempts to use a system on real problems, and that process is still unfinished. Therefore only the "kernel" part of the model will here be presented – those entities and relationships that are crucial for understanding the basic ideas. Figure 4 shows eight database tables and
relationships between their fields. Four entities make the kernel of the proposed model: design plan, design process step, action and document. The attributes for each of mentioned entities are stored in corresponding table. Each of these entities is categorized, having a list of its categories in separate table. The basic idea behind categorization is to enable implementation of different behaviour for various kinds of entity instances.

Table "design projects archive" contains main attributes of design plans. The data for both the skeletons and finished plans are stored in this table. The difference between various forms of plans is determined by the appropriate plan category. The task of program procedures and interfaces is to implement separate management and behaviour of skeleton plans, active plans and finished plans. Each kind of design plan contains a list of design process steps, whose attributes is stored in "design process steps" table.

Each design process step contains a list of actions, whose attributes are stored in "actions" table. "Step" and "action" represents two different levels of workflow nodes. The design process step is devised as being at higher (abstract) level of solving a subproblem or discussing an issue, while the action is more concrete designer activity like performing a calculation, modelling geometry, detailing, etc. To complete a design process step, a set of actions should be executed. As both steps and actions represent nodes in the design process workflow, they have almost the same sets of attributes.

During the work on the project the designer has to confirm the completeness of every step and action. More important, the designer has to write the rationale and any other significant data for every decision made in completed steps and actions. The sequence of steps is predefined in skeleton, but it could be changed, and new steps and actions could also be added. The system enables iterations within main project checkpoints – that means that every step and action could be repeated several times. The checkpoints are one of the steps' categories that include procedures for evaluation of the current project state in context of terms, expectations on solution quality, etc.

The process of using the proposed system when working on particular design project (figure 3):
- Choosing the appropriate skeleton, copying all the records of skeleton to new "active" plan for a particular project
- "Filling" the empty fields of "active" plan with

![Figure 4](image-url) The "kernel" part of design projects' repository prototype database structure
new knowledge, adding new steps or actions, recording the exact workflow sequence.

- Completing the project – the "active" plan becomes the "archived" plan, the records remain in the same table.

Every design step and action, together with all their attributes and links are being recorded. The sequence of execution is also completely recorded, so the whole project solving process history (including all iterations) could be restored. The rationale should be written in "memo" fields in "steps" and "actions" tables.

Each design plan, step and action has a list of links to corresponding documents. These documents either contain the knowledge needed for the particular step or action, or belong to the documentation set of the product being designed. Table "documents" contains basic data about all kinds of documents that could be linked to particular design plan, step or action. To which of previously mentioned entities a particular document belongs is indicated by filling one and only one of the three fields: "action ID", "step ID" or "design plan ID". The document management function of the proposed system overlaps with the functions of PDM systems. However, we think that it is necessary that a proposed system has an organized structured access to all the documents being used in the design process in order to build effective searching mechanisms for knowledge reuse.

In this research phase, there are still some open questions regarding recording and indexing of design rationale. Further research efforts should enrich the model with new entities, attributes and procedures for resolving these issues.

4. THE SOFTWARE COMPONENTS OF THE PROPOSED SYSTEM

Figure 5 explains the role of particular software components in the processes of design plans creation and execution. Operations of creating and executing the design plan are supported with "service" tools.

Figure 5  Roles of system elements in the usage process of the proposed system
and applications. These are very complex procedures as they must ensure the highest possible level of accuracy of the created plans. In the process of creating and executing of design plan, the designer must have on disposal all the available knowledge about software tools used in a particular design environment. Similarly, while planning or executing the design process, the designer should have interfaces to engineering databases (with product data), as well as to knowledge bases about design process and products being designed. Those interfaces should also be realized as “service applications”. The design plan creation starts from “empty” plan frameworks and process templates which consists of elementary entities and database structures.

The design process structure could be decomposed to a hierarchy of design tasks and subtasks. Therefore, it is not necessary to define a design plan at the highest (most complex) level of design task abstraction. Computer based support for complex design tasks can be modeled as a hierarchy of plans and subplans. The design process control at higher abstraction levels can then be left to the designer. Previously completed and tested plans should be stored in a separate database (archive) to be available for reuse or to be used as the subplans in a more complex plan being created.

5. ISSUES OF DESIGN REPOSITORY USAGE IN PRACTICE

An essential issue for the proposed system usage is to find a good balance between amount of information that a designer should record in system and the time he spends for this purpose. To achieve this goal, two components are of primary importance – interfaces and the product development ontology. A lot of research work and testing remains to be done to develop good and intuitive interfaces (and procedures) that would save designers' valuable time. The role of ontology is to provide design community with platform for better communication. There should be no misunderstandings about what is being recorded or what is being searched for.

The other question of practical usage is what types of knowledge and information the novice designer should be able to find in proposed system? According to Ahmed et al, (2003), the experienced designers were observed to follow certain design strategies which enabled them to evaluate their decisions prior to implementing them. Some of the questions from these strategies most properly describe what issues a “guiding” system should be able to resolve for the novice designers:

- What should be considered further down the design process? The skeleton of a design plan includes a list of further steps and actions down the design process. Novice designer is able to analyse the process of particular task solving.
- In which projects did similar issues arise and how were they resolved? Recognizing the similar issues in the archive of finished projects requires knowledge indexing system (and ontology usage) to able to find appropriate captured knowledge.
- What issues are relevant, and which are the most important? This part of experienced designers' knowledge should be written in explanations of actions and steps of design plan skeleton.
- How complete is the task, will further information be required? A special category of design plan steps should provide procedures for performing checkpoints and reviews. In design plan skeleton such step should contain a list of information required for analysis of project completeness.
- How much can I expect to achieve if I continue this approach? Maybe the most useful information captured in previous projects is why something gone wrong, why certain approach haven't reach the goal. Knowing this at the start of new project, the same situations could be avoided.

6. RESULTS AND KEY CONCLUSIONS

A specific design process has been analysed in order to develop and test the prototype of database structure (entity – relationship) model. The chosen design environment has well defined and organized design process that could be easily represented with proposed entities. Mentioned environment is a small design office carrying out projects for garden and building equipment manufacturer. The prototype relational database model, and a part of user interface is finished. Current work focuses on preparing examples of design plans as the basis for further development of system functionality, usability and efficiency. While developing the model structure, it becomes evident that a test implementation will require a lot of efforts in coordinating the perception of model elements and structure among potential users. The other important question is how to ensure the designers will write down the appropriate explanations (rationale) when they are necessary. One of the key requirements for the proposed system
is that it should take as little as possible of the designer's time. Previously mentioned facts affect many issues in preparing the terminology, the procedure of system usage and the interfaces. Another important issue for further development is the eventual transfer of the proposed model from cheap software environment to more powerful software and hardware environments, to be able to satisfy the requirements of larger design offices. Already at this phase of the project it showed up that a proposed system would generate a huge database in a relatively short time of usage. In the context of design information management, a design plan is the most complex object that integrates design information through sets of references to other (simpler) objects. Such an approach enables very complex queries and views of the single design project, as well as the set of similar projects. The main goal of the proposed system is to store the design rationale in the context of the design process history. In the next research phase we will try to find the mechanisms of indexing, searching and retrieving the stored rationale to be able to reuse it efficiently at the beginning of the new design projects.

ACKNOWLEDGMENTS

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 research is performed in co-operation with KLEX d.o.o., Zagreb, a mechanical engineering design office.

REFERENCES

Ahmed, S., (2005) "Encouraging reuse of design knowledge: a method to index knowledge", in Design Studies – article in press


