DESIGN INTENT MANAGEMENT FOR DESIGN REUSE

J.S.M. Vergeest, Y. Song, T.R. Langerak

Delft University of Technology
Faculty of Industrial Design Engineering
j.s.m.vergeest@tudelft.nl

Keywords: Reverse engineering, reverse design, redesign, design intent, shape reuse

Abstract: *The creation of new designs depends on precedent designs. There is an extreme variety in the types and degrees of dependence, ranging from direct copying and reverse engineering till unconscious inspiration of the designer by existing products, earlier experiences or education. To increase the competitiveness of design and engineering, the reuse of precedent work can be systematically incorporated into the design process, e.g. by establishing a methodology around product platforms and reconfigurable product architectures. A question is whether design reuse can happen without organizational investment and without having to anticipate on what types of precedent work is prone to reuse. Ultimately, the designer should be able to arbitrarily select a portion from an earlier design and insert that portion into the current design, where the imported information is expected to "adapt" to the context of the new design. We have investigated these questions in the domain of freeform designs, where the product's shape should fulfill aesthetic, ergonomic and functional requirements. In this paper we describe a framework for design intent management and propose a procedure to obtain a freeform shape parameterization according to an emerging design intent. The method involves feature recognition in the freeform domain, and the availing of shape parameters to the designer. We have partially implemented the method in a CAD environment and successfully tested it for several types of freeform features.*

1. INTRODUCTION

Design reuse happens in every design project, explicitly or implicitly. Design reuse is can be observed in various manifestation forms. 1) Any decision or action in a design process may be inspired from what the designer remembers, learned or experienced in the past. 2) The designer consults an expert about an issue and gets informed how the expert solved the problem in a comparable situation. 3) The designer searches in archives for precedent designs and products in order to gain knowledge or inspiration related to the design problem at hand. 4) Structures and/or shapes from previous design documents are applied, copied or otherwise reused in the current design. 5) The current design is a modification of a previous design. These are just examples of possible forms of design reuse.

It is generally recognized that reuse in design is not only a naturally occurring phenomenon, but also a significant profit factor, increasing the performance of design [1] [2]. On the other hand, it appears to be difficult to apply design reuse at the document (or representation) level. Reuse at the document level is the insertion of a portion of an already existing design document into the current design document. In CAD environments, where a document is a digital representation of the design, this issue is most obvious. Only in very well-defined contexts it is feasible and useful to copy and paste a part of a CAD model into another CAD model. If the design context is built around a product family with a defined modularity, or platform-based, then indeed, parts, components and modules become sharable among different designs [3]. However, the reuse of a design portion on an *ad hoc* basis is practically unsupported yet [4].

As a simple example, suppose that a designer observes a small imprint in the shape of an existing product, and he/she wants to use that imprint in a new CAD-based design. It will be relatively easy to extract the surface geometry containing the imprint. However, when that surface region is inserted into the shape of the new design, it will probably not smoothly fit with the surrounding geometry. From the designer's perspective, this is an unnatural setback, and any work to smoothen the insertion would disrupt the "natural" design flow. Recently, in [5] a method to automate the adaptation of inserted freeform features was presented. After copy/pasting and (automatic) adaptation of a freeform feature, further adjustments and modifications are generally needed. In case of the imprint, its height or depth may need to be adjusted, or *e.g.* in case an entire styling line is copied, the characteristics of its cross section profile, or the general flow of the styling line, without affecting the cross section profile could be subject of change. We can now distinguish two circumstances with very different prospects of modifying a reused feature. In case I the designer can achieve the intended modification by operating shape handles already available from the original design. Which shape handles are available depends on how the feature was originally created, and which CAD tools were used. If, for example, the feature reused already has an associated parameter controlling the sharpness of a cross section profile, and modifying this sharpness is what the designer wants, the process of reuse and modification will be very efficient. In case II the shape handles to directly achieve the intended modification are not available. The designer then has to rely on general tools provided by the CAD system at hand, which will typically work on low-level geometric elements and the process will be far from efficient.

Case I will be much less frequent than case II and as a result the designer will often be disappointed by the seemingly static nature of the reused portion of the design. There is thus a discrepancy between the expectations of the designer, in terms of what he/she intends to achieve with the reused portion on the one hand, and what is supported on the level of digital representations. It should also be mentioned that case II includes situations in which parameters and shape handles are totally absent, as for example when geometry is imported using 3D scanning, resulting in a large set of data points without any means of high-level modification.

In this paper we present a model to describe the process of design reuse on the level of design intent and on the level of document and representation. We will show that successful design reuse depends on how well design intent can be made explicit and managed. In the domain of freeform shape features we present a method and concrete examples of design reuse.

2. A MODEL OF THE DESIGN REUSE PROCESS

To discuss the process of design reuse we will need to consider three levels of product manifestation,

1. The mental level, including images, structures and other representations of intents that a person may have in mind concerning a design.
2. The level of concrete documents. It covers externalized designs such as sketches, drawings, texts, paper or clay models, as well as digital representations of them, including CAD models.
3. The physical level, including manufactured products, prototypes and physical objects in general.

We emphasize that the purpose of this division into three levels is not to propose a new psychological insight of the design process but to clarify the contrast between the levels in terms of the flexibility with which intentions and representations can be changed and adapted. A designer reportedly can associate mental and concrete models from the past with mental models of a current design project easily, and he/she can communicate these intents to other people accurately. However, to materialize these associations and intents appears very hard to achieve on the concrete level.

To visualize actions such as mental model building, externalization and design reuse we define three entities according to the aforementioned levels, 1) mental design model or design intent, 2) concrete design model, document or representation and 3) physical object, see Figure 1. Obviously, the categorization of things towards the three levels is not always strict. For example, a simple silhouette cut out of a piece of card board could be regarded as a representation form of a design but also as a physical object. In the course of this paper, level 2 will typically refer to a digital model (CAD model) of a design.

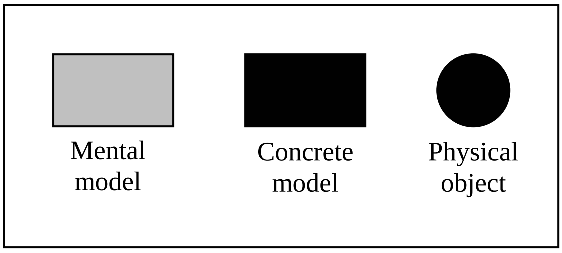


Fig. 1. *Three levels of occurrence.*

Using the three entities of Figure 1, some basic actions of design can be depicted in a very simple, perhaps oversimplified way, but it will serve our purpose.

Externalization can be regarded as the transfer of information from level 1 to level 2; the concrete model (*e.g.* a digital representation) is being extended, see Figure 2. We will refer to this type of action as action *A*. Another interpretation of action *A* is realization of a design intent.

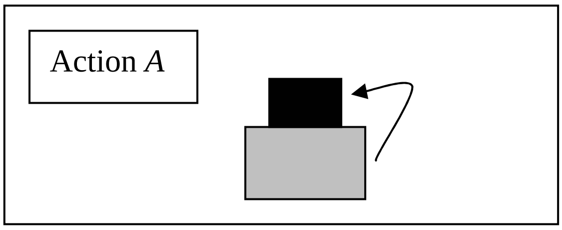


Fig. 2. *Externalization is grow of the concrete model by feeding from mental model (type of action A).*

The second type action that we distinguish is action *B*, the influence on a mental model by physical objects. This type of action can be said to take place when somebody is inspired by an existing product or piece of art or by nature. Then it can be said that a mental model emerges or is extended due to that object. Figure 3 depicts action *B* symbolically.

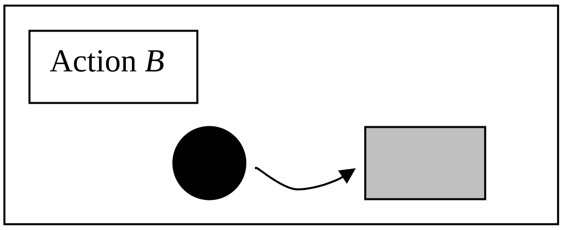


Fig. 3. *Physical objects give rise to* (*contribute to*) a *mental model (type of action B).*

Action *C* (Figure 4) is the back-influence on the mental model by the belonging representation. It can be understood as the inverse of process *A*. The processes *A* and *C* are sometimes collectively referred to as a dialectics [6] in design.

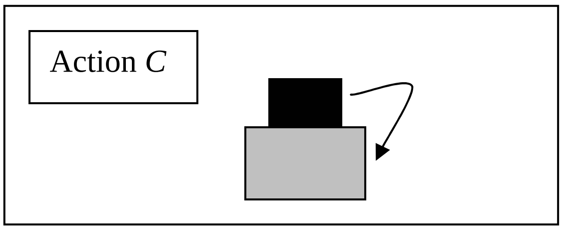


Fig. 4. *A concrete representation-in-progress influences the designer's mental model (type of action C).*

Transforming a concrete design representation into a physical object (type of action *D*, see Figure 5) is typically a final step in the process, and refers to manufacturing or rapid prototyping. The absence of the mental model in Figure 5 symbolizes the "automated" character of the action; it is supposed not to be influenced anymore by the designer (which is, of course an oversimplified picture, but it serving the purpose of our discussion).

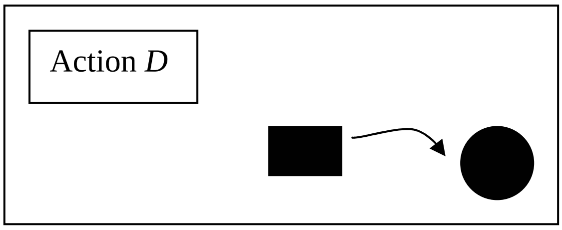


Fig. 5. *Transformation of a concrete design representation into a physical object (type of action D).*

Finally, the creation of a design representation from a physical object is action *E* (see Figure 6). This process is commonly referred to as reverse engineering, and 3D surface scanning is a technology to support this type of action. Theoretically this action can be automatic and for that reason no mental model is included in Figure 6. However, in practice, a reverse engineering process requires the involvement of a designer who controls the process based on mental models that he/she creates, but we leave out that aspect for the moment.

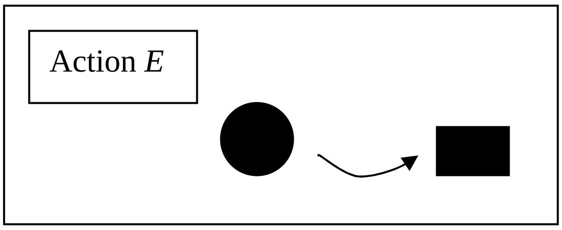


Fig. 6. *Creation of (contribution to) a design representation from an existing physical object (type of action E).*

In an extremely simplified way, a design process can be depicted in expression (1) as sequence of actions.

$$B\ A\ C\ A\ C\ A\ C\ B\ A\ C\ D. \qquad (1)$$

The first "action" (*B*) of process (1) is the creation of a mental model inspired by some physical object, where no concrete representation is yet existing. Then, alternating, actions *A* and *C* cause a continuous building of both a mental model and a concrete representation of the design. Action *B* reoccurs as to enrich the designer's mental model, and after one more iteration (*A* and *C*) the concrete representation is completed and physical objects are derived from it in a manufacturing facility (action *D*). The actions *B* and *C* extend the mental design model, whereas action *A* extends the concrete representation. Therefore in (1), the influence by a physical object on a concrete representation is indirect; it occurs through the designer's mental model. The substring *"B A"* can represent design reuse as well. in each of the two occurrences in expression (1), see Figure 7. For the substring *"B A"* in the very beginning of (1), the designer generates a mental image of a design based on a physical object and then starts to build a concrete representation of the design. The second occurrence of string "*B A"* in (1) can represent the following situation. Prior to "*B A"* both a mental model and a concrete representation of the design exist. The designer observes a physical object and wants to include a part or an aspect of that object into the new design. This design intent is an extension of the mental model (action *B*), and subsequently the concrete representation is extended accordingly (action *A*).

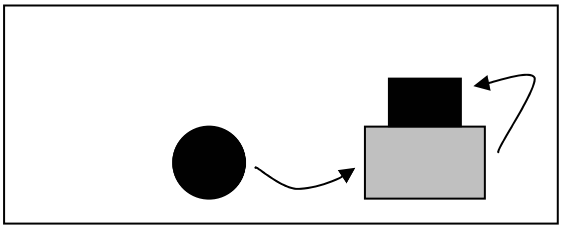


Fig. 7 *Design reuse through the mental model. Action B and followed by A, where both mental and concrete model exist.*

As mentioned before, design reuse is a very frequent action, and it may occur implicitly and unconsciously. However, design reuse proceeds predominantly *through the designer's mental model*. Although design reuse directly, on the level of concrete representation seems much more efficient, it is observed rarely [4]. Design reuse on the concrete level could occur in two ways in process (2), each represented by the substring "*E C*".

$$E\ C\ A\ C\ A\ C\ E\ C\ A\ C\ D. \qquad (2)$$

The first occurrence in (2) of *"E C"* can represent a CAD-supported reverse engineering process, where *E* is the creation of a digital representation from an existing object or product, followed by creation of a mental model, see Figure 8.

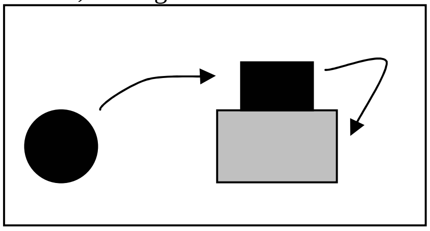


Fig. 8. *Design reuse through the concrete representation. Action E (where no mental model exists yet), followed by action C,*

The second occurrence of *"E C"* refers to a situation in which both a mental and a concrete model have developed, and where the concrete representation is directly influenced by a physical object. An example of such a situation is, again, reverse engineering, where 3D scan data from a physical product is inserted into the CAD representation. Once the concrete representation has changed, the mental model will change as well, in part due to the designer's reflection on what happened to the concrete representation (downward arrow in Figure 8).

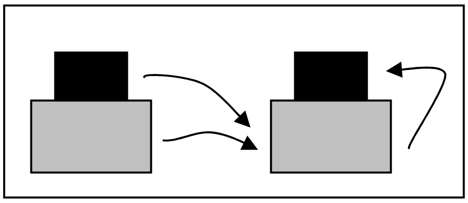


Fig. 9. *Mental model is influenced by a mental model or by a concrete representation of another design project.*

The actions *A* and *C* can be seen as the mutual influence of a concrete and an abstract model within a single design process.

Two different design processes can influence each other, as shown in Figure 9. We will refer to these as "the first" and "the second" design process, carried out by the first and the second designer. The arrow from mental model of the first design process to the mental model of the second symbolizes a communication between two designers, whereas the arrow from concrete model to mental model refers to inspiration of the designer of the second process by the representation obtained in the first one. Obviously, the design processes need not to happen simultaneously; if a history of the first design process was stored, then intermediate representations can be retrieved and be used as inspiration for the second process. The arrow from mental model to concrete representation (action *A*) within the second design process in Figure 9 is drawn for completeness only. It indicates that after information flow from one design process to the other, the "common" process is resumed.

The most challenging type of design reuse is the one on the level of concrete representations (or document level), see Figure 10. The arrow from left to right denotes the transfer of model data from the first to the second design process. The central issue here is that in this transfer the mental model (and hence the designer) is not explicitly involved. The data transfer is in favor of the second design process and as such typically initiated by the second designer, who has, however, no direct influence on the changes to the concrete model. In terms of a practical case, the CAD representation of the second design process is modified due to data from another digital representation. Although the digital representation of the second design process is thus enriched, two types of defects may occur: 1) the imported data does not fit correctly into the digital model of the second design process, for example a piece of surface copied from the first design may appear not to connect smoothly into the surface model of the second design, 2) the design process after importing the data may be severely hampered due to loss of design intent, for example, the geometry of a parametric feature is correctly copied into the digital model of the second design process, but the parameters themselves cease to function. After the data transfer, the second designer interprets the change applied to the concrete representation, shown in Figure 10 as an action of type *C*. Commonly the then next action will be of type *A*, which is the further development of the concrete model from information at the mental level.

Failing design reuse has two downsides

- Loss of time and effort to repair the defects. In the example mentioned above, the designer may spend some time to reshape the imported surface in order to connect it smoothly to the embedding geometry. In the case of lost design intent, the designer may attempt to recreate the parameters or other operators that were lost.
- A potential of improved design efficiency is unused if design reuse has its bad reputation among practical users. A methodology based on design reuse will not develop, and instead avoidable repetition of work will occur.

Failing design reuse can remain unnoticed until at a late stage of the design process, so that the cost of repair and work-around gets even higher.

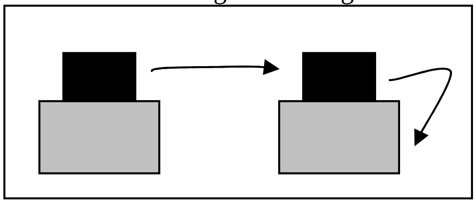


Fig 10. *Information transfer on the level of concrete representations, followed by action C.*

As mentioned, if design reuse is explicitly organized around modules and platforms, it will function efficiently. In some areas of design, for example freeform styling, it is not obvious how to systematize design reuse. If we still want to profit from designs at the document level then 1) designers should be able to know *in advance* whether reuse is feasible in a particular case and 2) techniques to facilitate incidental design reuse should be available. In the remainder of this article we will focus on these two issues and provide some practical solutions in the area of freeform surface design and shape styling.

3. DESIGN INTENT AND DESIGN REUSE

Freeform surface design and shape styling are typically not practiced in a systems-design fashion. Characteristic of freeform modeling is the usage of many different shape modeling (computer-based) tools on various levels of detail. Detailed local surface tweaking can be followed by globally bending an entire object. Numerical parameters are not often applied although the concept of continuous modification of local and global properties is apparent in shape design [8]. It has been observed that a series of detail actions served to reach a goal which was clearly definable by the designer as a design intent, but could only be achieved by low-level actions [9]. In such a case, the designer's intent *is not included* in the concrete representation, although apparent in, for example, the graphical presentation of the design. To explicitly include the intents of a surface design into the digital model, freeform features should be available, and be applied by the designer. Basic properties of shape, such as key dimensions and edge roundness should be directly editable, even when the numerical values of the parameters are not (yet) of importance. What now can happen (referring back to Figure 10) is the following:

- The first designer has created a concrete model using low-level tools, although the resulting design seems to contain freeform features. However, no parameters were ever introduced on the concrete level. On the mental level of the first design freeform features may or may not exist.
- The second designer notices the shape obtained in the first design by means of the concrete model. He/she decides to reuse a perceived shape feature seemingly included in the model.
- Shape data is transferred from first to second concrete model.
- The second designer wants to adapt a feature parameter, but the parameter is unavailable since it was never defined.

Even if a parameter would have been defined by the first designer, and if the parameter definition (or design intent) were transferred to the second design process too, it could still be the case that the second designer wants to adjust a different parameter than the one available.

The scenario just sketched calls, of course, for a communication on the mental level (Figure 9), in which the first and second designer gain understanding about each other's intents and way of building their concrete models. However, for various practical reasons the mental model of the first designer may be unavailable. So we still need a methodology in which design reuse happens at the document level (Figure 10), where we consider the following possible requirements:

1. If the correct design intent from the first design process is represented by the received model data, it should be applicable efficiently in the second design process.
2. If no design intent is communicated the second designer should be able to "create" an intent, in order to proceed with the design in an efficient way.
3. If the wrong design intent is communicated it should be replaced by the appropriate, efficient tools.

Here it should be understood that the term intent refers to tools (or CAD functions) on the document level. For example, in case of a feature-based design, the design intent is reflected by the parameters and constraints included in the concrete model; these parameters and constraints define the modes of operation (*i.e.* tools) of the concrete model.

Requirement (1) is met when both in the first and in the second design process feature-based models are built. In the domain of prismatic and analytical shapes, fully parameterized models are possible with state-of-the-art CAD systems. In that domain the exchange of (sub)features among feature-based systems is achievable. In the freeform domain, however, parameter-based or feature-based methods are still topics of research [9], which means that requirement (2) is actual. The same holds even for the prismatic/analytical domain in case (a part of) the models are built using low-level geometric elements. The receiver of the data may expect to be able to modify the height of a box using a parameter, while such parameter was never defined and, instead, the box was created as an unordered set of planar sections. What will happen is that the second designer contemplates about the received concrete model and start to adapt his/her perception about the structure of the model, or in other words, the mental model is influenced by the concrete model (downward arrow, action of type *C* in Figure 10). The second designer will adapt his/her understanding of the model and generate new intents. In the example of the box the designer may prepare to modify the planar sections individually. Alternatively, he/she may attempt to redefine the model into a parameter-based feature model and effectuate the modification using the parameters. Finally, requirement (3) implies the ability to delete parameters, constraints and tools that were defined according to a design intent, and to replace them with new ones.

The creation, deletion and replacement of parameters, constraints and tools on the concrete level is equivalent with design intent change on the mental level. The philosophy is that design intents naturally develop during design processes and should not be hindered by unexpected properties in the concrete models. We have seen that during design reuse there is a risk that design intents get lost. In the ideal case the computer should be able to avoid the loss of design intent by automatic recognition methods on the concrete level. Feature recognition has been proposed for that purpose in [10]. Indeed, from a set of planar sections, a parametric boundary representation of a box can be reconstructed, and be availed to the user for parametric design actions. It remains a problem how to deal with the non-uniqueness of a design intent for a given concrete model. For example, a rectangular box can be represented by multiple choices of parameters. And in some cases the design intent is equivalent with *no parameters at all*. So, even a computer that is able to translate concrete models into other concrete models according to a design intent sill needs information about the intent itself. In typical feature recognition systems this has been solved by the introduction of a library of feature classes, which serves as a superset of representations of design intents, where for practical reasons the library is restricted to frequently occurring feature classes. The success of design reuse then depends on the richness and completeness of the library.

In the domain of freeform styling the definition of feature classes and libraries is known to be hard [11]. Recent studies suggest that a mix of automation and involvement of the designer in the definition of feature classes is more productive than any fully automatic feature recognition system based on a static predefined feature class library. In [12] a method is developed in which the user can edit a styling line in a freeform surface using high-level shape tools such as height and width of the cross section profile. However, there is no such library containing all possible types of styling lines, since the number of types would be uncountable. Instead, the user specifies a simple template to characterize the feature, and the system then initiates a recognition and localization process on order to generate the high-level tools needed for subsequent editing. Referring to Figure 10 this methodology of design reuse requires some effort from the second designer, who should "understand" the design intent of the first designer and translate it into his/her own design intent. This requires design intent management on the mental level, as well as feature class definition on the concrete level. The further actions of the reuse process can be automated as will be presented below.

4. APPLICATION EXAMPLES IN FREEFORM STYLING

In section 3 we discussed the three possible requirements (conditions) of design reuse, namely explicit transfer of design intent, no transfer of design intent, transfer of a wrong design intent. We also concluded that in the freeform domain, condition 1 cannot be supported, not even with advanced techniques. Therefore, here we look at the possibilities to support requirement 2. If requirement 2 is fulfilled, then requirement 3 can be met as well, in the understanding that a wrong design intent can be removed; then the model can be considered as containing no design intent, which is condition 2.

We illustrate two ways of freeform shape reuse. The first is practically fully automatic and relies on predefined feature classes. The second method requires some explicit information about design intent from the designer.

Predefined feature classes

Pioneering work on freeform recognition and manipulation has been carried out by Spanjaard [13] and Song [14].

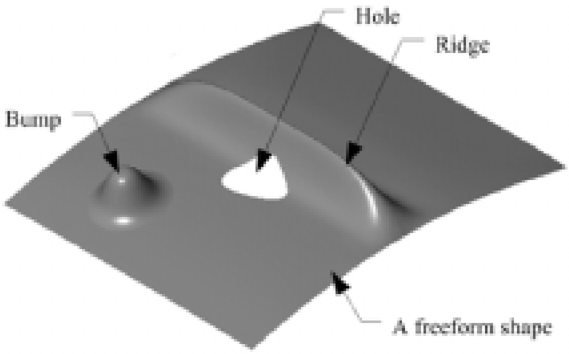


Fig. 11. *Example of three freeform features (hole, bump, ridge).*

Using a limited set of three predefined feature classes (hole, bump, ridge, see Figure 11), requirement 2 can be met as long as the problem remains within the domain of the three types of freeform feature. On the receiver's side of the data transfer (Figure 10), the original design intent can be automatically detected and any freeform feature occurring in the transferred data is recognized and availed explicitly to the user, including the parameters belonging to the feature, see Figure 12.

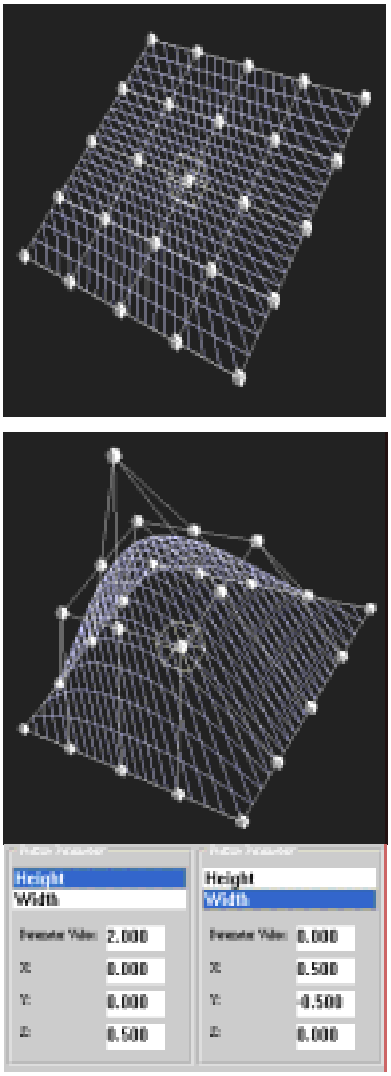
Fig. 12. *The height of a bump is edited according to the original intent.*

In the reuse process a bump imprinted with a text was imported. Although the representation was a dense surface mesh, the designer wants to be able to adjust the height of the bump directly, rather than manipulating individual surface facets. This support could be provided based on fitting techniques described in [13]. Thus, the intent from the first designer is made available to the second designer on the concrete level.

User-defined feature classes

If the transferred data appears to contain a feature which is not consistent with any of the predefined classes, the method just described will fail. There are at least two ways to solve this. The first is to extend the feature class library. The drawback of library extension is that it relies on the involvement of the experts who develop and maintain the software system. And, if the library contains many classes, the computational effort of feature recognition will increase, since the data will have to be compared against a larger number of feature templates.

Another approach is to involve the user in the definition of the feature class. In [11] a system is proposed that allows the user quickly specify a feature template, which is subsequently used in the feature recognition process, see Figure 13.



2.000
X: 0.000
Y: 0.000
Z: 0.500

0.000
X: 0.500
Y: -0.500
Z: 0.000

Fig. 13. *The user defines a freeform feature template.*

Once the feature class has been defined, the user has the option to include it to the library permanently.

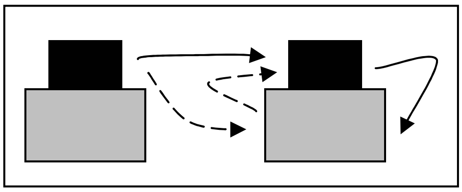


Figure 14. *Concrete model of first design is (partly) reused for the second design. The mental model of the second designer is involved in the communication as well.*

The method is depicted in Figure 14. The main information stream is (like in Figure 10) on the concrete level, from first to second design. Designer 2 interprets the concrete model from design 1 (dashed arrow from left to right) and influences the concrete model (action of type *A*, dashed upward arrow). From then on the "regular" design process proceeds with action of type *C* (downward solid arrow). Clearly, the model in Figure 14 is a combination of the models in Figures 9 and 10.

5. CONCLUSION

Reuse of previous design work is both advantageous and frequently occurring. Most of the reuse processes proceed on the mental level. To effectuate the design on the concrete level, support is severely lacking. Fully automatic reuse processes have been developed but are limited to a predefined set of possible design intents or features. This is true in every geometric domain, but even more in the domain of freeform shapes. A feasible way to overcome the limitations is to let the designer indicate which intents (and thus features and parameters) are needed for a particular reuse process. To further support and streamline design reuse, design intents should be represented explicitly, in a CAD system-independent manner. Research in this direction is currently in progress.

References

[1] A.H.B. Duffy and A.F. Ferns, "An analysis of design reuse benefits". U. Lindemann *et al.*, (Eds.), proceedings of the ICED99 Conference, Technische Universität München, 1999, pp 799-804.

[2] S.N. Smyth and D.R. Wallace, "Towards the synthesis of aesthetic product form". Proc. Proc. DETC2000/DTM-14554,, ASME, New York, 2000.

[3] A.P. Hofer and J.I.M. Halman, "The potential of layout platforms for modular complex products and systems". In: I. Horváth and P. Xirouchakis (Eds.), Proc. TMCE 2004, Millpress, Rotterdam, 2004, pp 573-584.

[4] J.S.M. Vergeest, I. Horváth, S. Spanjaard, "A methodology for reusing freeform shape content". Proc. of the 2001 Design Theory and Methodology Conference, DETC'01/DTM-21708, ASME, New York, 2001.

[5] C. Wang, J.S.M. Vergeest and P.J. Stappers. "Design reuse in product shape modeling: global and local shape reuse". WSEAS transactions on information science and applications, 1(1), 2004, 332-340.

[6] G. Goldschmidt, "The dialectics of sketching". Creativity research Journal 4 (2), 1991, pp 123-143.

[7] J.S.M. Vergeest, I. Horváth and S. Spanjaard, "Parameterization of freeform features". In: A. Pasko and M. Spagnuolo (Eds.), Proc. Shape Modeling International, IMA-CNR, IEEE, Piscataway, 2001, pp 20-29.

[8] R. Dumitrescu,, J.S.M. Vergeest and T. Wiegers, Analysis of a freeform shape modelling process. In Aoki, H (Ed.), 6th Asian design international conference. Journal of the asian design international conference (pp. 1-8). Tsukuba: Institute of Art and Design, Univ. of Tsukuba, 2003.

[9] P. Nyirenda, M. Mulbagan and W.F. Bronsvoort, "Definition of freeform feature classes", Proc CAD 2006 Conference (in press).

[10] J.J. Shah and M. Mäntylä, "Parametric and feature-based CAD/CAM". John Wiley & Sons Inc, New York, 1995.

[11] T.R. Langerak, J.S.M. Vergeest, "A new framework for defining free form features and its application on feature composition". Proc of the ASME DETC 2006 Conference (in press).

[12] T.R. Langerak, TR, J.S.M. Vergeest, Y. Song, "Parameterising styling lines for reverse design using free form shape analysis". In Proceedings of IDECT/CIE 2005 (pp. 1-8). New York: ASME.

[13] S. Spanjaard and J.S.M. Vergeest, "Comparing different fitting strategies for matching two 3D point sets using a multivariable minimizer". D.T. Mook and B. Balachandran (Eds), 2001 ASME design engineering technical conference and computers and information in engineering conference, ASME, New York.

[14] Y. Song, J.S.M. Vergeest and W.F. Bronsvoort, "Fitting and manipulating freeform shapes using templates". Journal of Computing and Information Science in Engineering, 5 (2), 2005, pp 86-94.