INTERNATIONAL DESIGN CONFERENCE - DESIGN 2002
Dubrovnik, May 14 - 17, 2002.

PRODUCT MODELLING IN EARLY PHASES OF THE DESIGN PROCESS

Tapio Korpela

Keywords: design methodology, requirements modelling, theory of technical system

1. Introduction

A technical process is needed to transform something from an unsatisfactory state to a satisfactory state. In this process human beings, technical products and an active environment co-operate. Some examples of these transformation processes and products are:

> *Scissors*: A whole sheet of paper -> paper divided into two pieces of paper,
> *Mobile phone*: An urgent need to contact a distant person -> A person is reached for a conversation or a message is left for her or him.

At first there exist only images of how to proceed a bad unwanted state of something to a desired state. Images may arise from an older product for the same purpose or from associations to products which have same kind of functions which are needed in the product to be created. Before a prototype we have only desired behaviour of the product (functions) and after the prototype has been produced, the real behaviour of the new product can be tested. In my paper I am going to present and discuss ideas on formal methods for modelling the product during the definition and the specification phases.

2. The chain from functions to solution principles

If an old product is a starting point to a new development, the chain from properties of an old product to the structural features can be revealed. This is a reverse engineering process compared with an original design process. Physical effects, geometric and material characteristics are the things which are tried to find out behind the properties. Designers apply their learned knowledge when they analyze these existing structures. The designer has to understand entirely these chains in order to develop the better solution.

2.1 The analysis process of an existing structure

The analysis process of a well known rotating shaving head developed by Philips company might look like as it is presented in the following figures. The rotating cutter and the assembly of a cutter and guard are presented in the figure 2.
The first electric shaver, using rotating cutters instead of the reciprocating cutters, was introduced to the European marketplace in 1939 just when World War II was beginning in Europe. The company was Philips in Holland and the structure was developed by Professor Alexandre Horowitz (1904-1982) [http://iavbbs.com/gflinn/biohor].

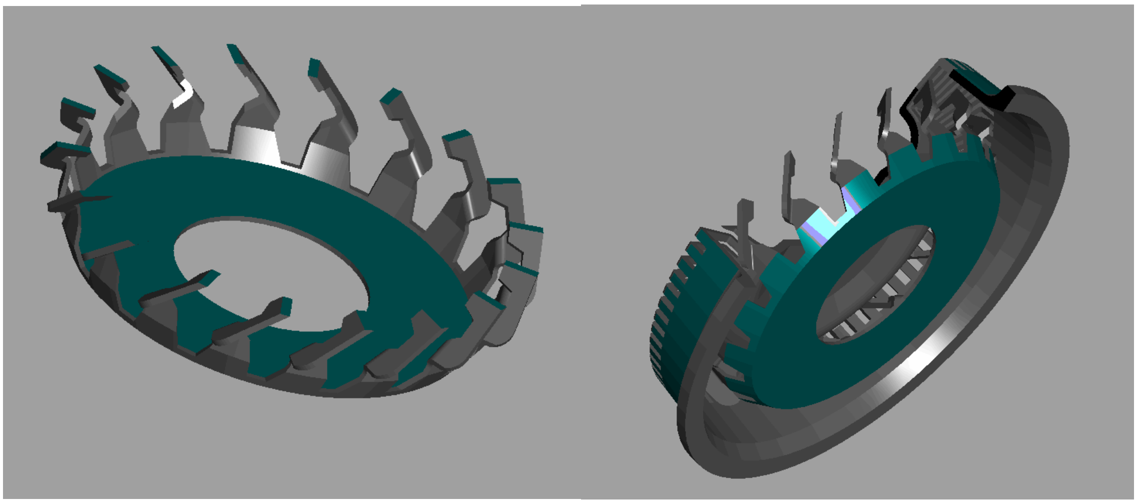

Figure 2. The cutter and the shaving head assembly

The cutting operation of a single hair can be imagined to happen as it is presented in the following sketch.

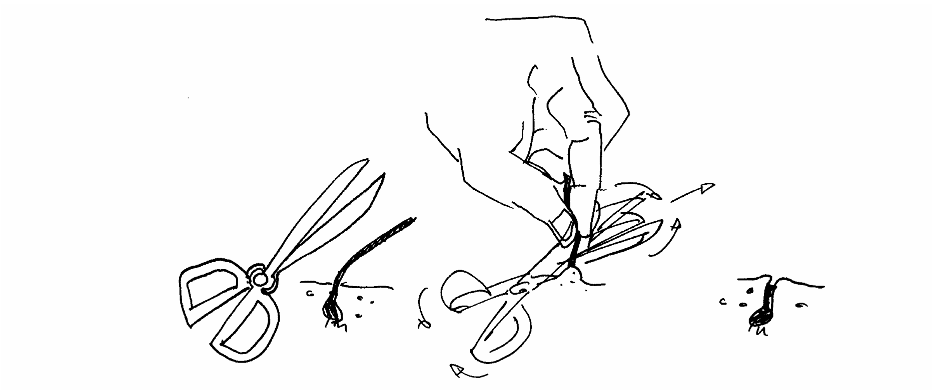

Figure 3. A preliminary sketch of the shaving operation

The closer examination of the shaving head reveals, that about twenty pair of scissors are functioned on the circle of the cutting head, when the cutter rotates against the guard. On the surface of the cutter there is assembled also a thin flexible plade, which is assumed to lift the hair before the cutting operation. The solution principle for the shaving head is presented in the figure 4.

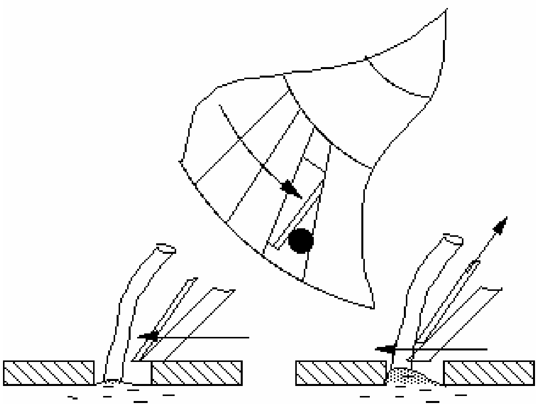

Figure 4. The solution principle of the shaving head

The causality from the functions of an old product to the structural features have been found out and these attributes can be the starting points for the new and better solution.

3. Theory of technical systems by Hubka and Eder

The design process of a totally new product usually starts from a quite abstract description of the problem and leads to a concrete solution. The designer tries to find economical means of achieving the desired properties in the product.The main phases in this process are clarification of the task, conceptual design, embodiment design and detail design [Pahl & Beitz 1986].
Vladimir Hubka and W. Ernst Eder describes a total concept theory for engineering design in their book: "Theory of Technical System" [Hubka & Eder 1988]. In this theory the product to be dersigned (technical system) is one part of the transformation system, in which the technical process is a formalized transformation process, which changes the state of operands in desired way. Transformation system consists of humans, environment and technical system, which co-operate with one another as well as with the technical process. The simplified model of the transformation system according to [Hubka & Eder 1988] is presented in the figure 5.

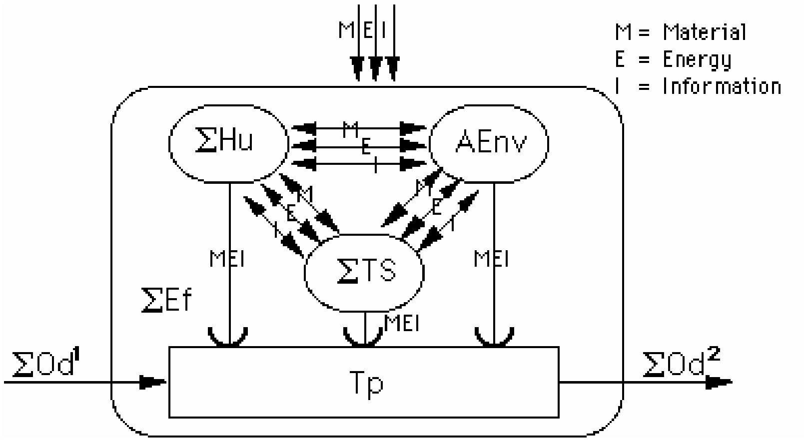

Figure 5. The simplified model of the transformation system [Hubka & Eder 1988]

Technical process (Tp in the figure 5.) is decomposed into the partial processes, which are related to each other by the states of operands between consecutive processes. Transformations in these partial processes are realized by certain effects, which are the means of achieving the desired transformation on operands. The effects are produced by the operators, which are: humans (Hu in the figure 5.), technical systems (TS in the figure 5.) and active environment (AEnv in the figure 5.).
The active units, which create the necessary effects in the technical system are called organs in the Domain Theory (DT) by [Andreasen 1998]. Parts or/and components maintain the organs in the final macine structure.
The Domain Theory by [Andreasen 1998] conciders the machine system from three different views, which are: transformation system, organ system and parts system. The concideration leads to a genetic model for the results of the design task, the chromosome model. The theory is also called "Artefact theory" [Andreasen 1991].

3.1 An example of modelling the technical process of cooking a cup of cocoa

For example if we have to make a cup of cocoa. In initial state we have cocoa powder, milk liquid and fine sugar. After the process we have a hot and sweet cup of cocoa. The transformation process is marked with a filled arrow from initial state to the desired state in the next figure 6.
The needed process can be read from the side of the cocoa box (magnifying glass in the figure) under the heading text: "Directions for use" The text is following:

> *Put one teaspoonful of cocoa and two of sugar in a cup and mix together. Add boiling milk or water and stir continously.*

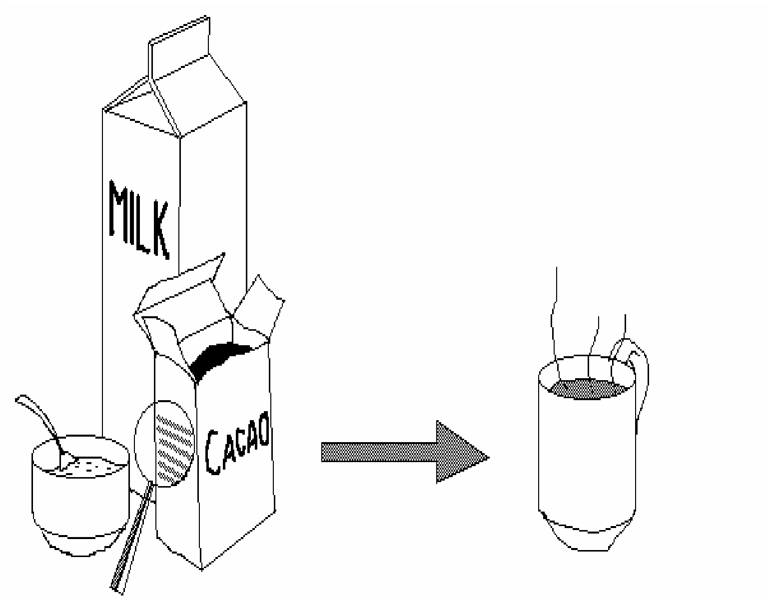

Figure 6. Transformation process to make cocoa

This scenario is a good starting point for the transformation modelling and the process can be divided into next subprocesses presented in the figure 7.

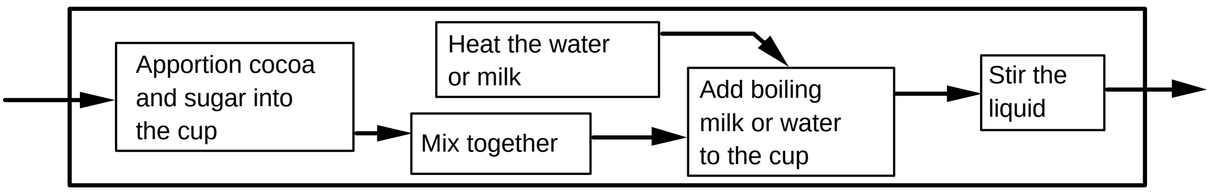

Figure 7. The technical process for the cooking of a cup of cocoa

If the cup of cocoa is going to be made manually in home kitchen, the equipments which are needed for this process are simple and obvious. But, if these manual operations are automated and included into the cocoa maker, the organs and structures are quite different. The technical system is now a self-service cocoa maker maintained by the bartender and used by the client as a self-service machine in a cafe. The initial technical process is presented in the figure 7. The first image of the system might look like it is presented in the figure 8.

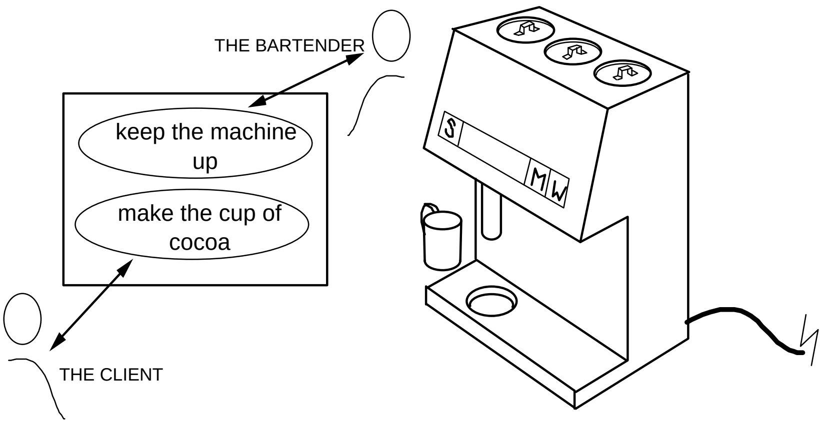

Figure 8. The first image of the system and the main users with their using processes

The interface of the system consists of the start button marked with a letter “S”. There are also the selection buttons for milk (marked with “M”) and for water (marked with “W”). The scenario for the client’s using process, which is also the directions for use could consists of the following lines:

1. *Put an empty cup under the tube.*
2. *Press the start button marked with “S” in the panel.*
3. *Wait until the milk and water butons marked with “M” and “W” start to blink in the panel.*
4. *Select the liquid into your cocoa by pressing milk or water button in the panel.*
5. *Wait until the lights in the buttons are gone out.*
6. *Take the cup out of the machine table.*

The function of use case -modelling is to find objects and behaviour of the system from the point of the user. The system is defined externally (look and feel) and the first draft of the user manual is created. If we now model the states of the cocoa machine during the cocoa making according to [Shlaer & Mellor 1992], it would look as presented in the next figure 9.

State transition model models the behaviour pattern of one object. Actions assosiated with the states gives the functions of the objects. Besides this objects have static relationships and they communicate with each other. Typical static relationship among objects can be aggregation (part of) or inheritance (kind of). The water tank and the heater element are parts of the cocoa maker. Start button is a kind of push button. [Shlaer & Mellor 1992] models event communication between state models and external entities such as operators (users of the system), physical devices, and objects in other subsystems by the object communication model (OCM).

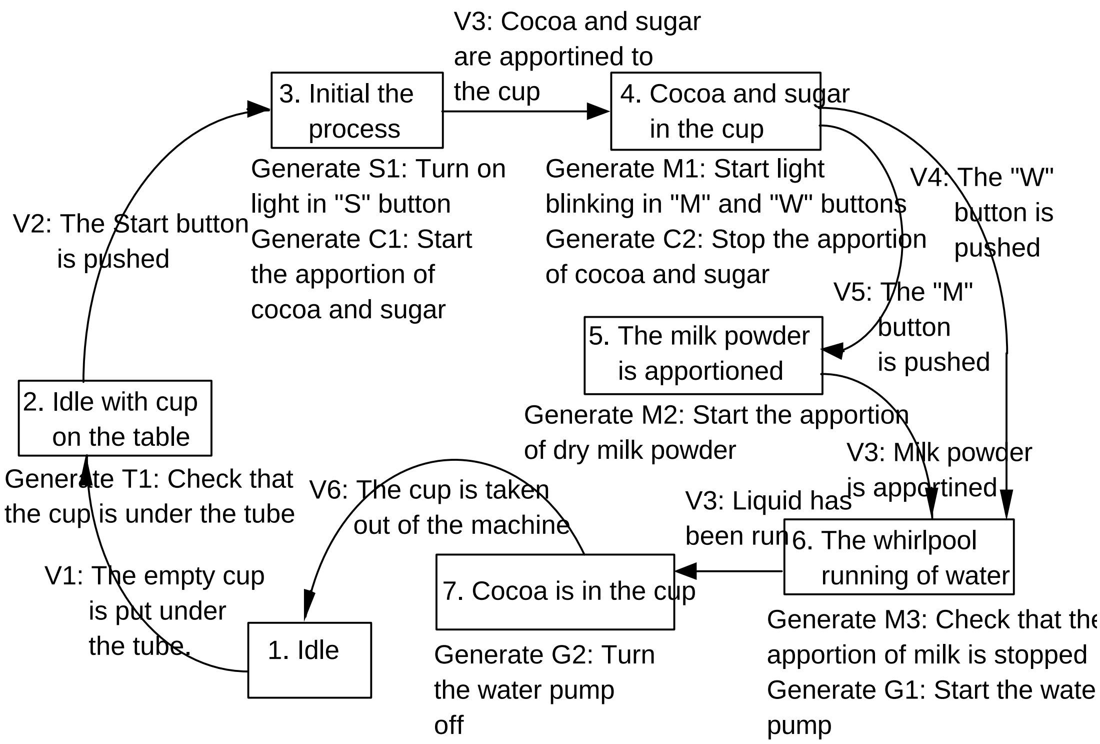

Figure 9. State model of the cocoa making process

4. Defining the product by using object-oriented analysis

In the previous modelling process of a cocoa maker an object-oriented analysis (OOA) method have been applied. An OOA process should be customer-oriented and implementation-free as far as possible. Therefore the modelling process starts by formalizing the customer needs in terms of requirement models. Jacobson divides the requirement model into a domain object model, a use case model and interfaces. Modelling the domain objects is a bottom-up approach, and when the use cases and interfaces are defined we have a top-down approach. The OOA process for software engineering,

according to [Jacobson et al. 1992], is presented in the figure 10.
Firstly, we can perceive tangible objects, which have direct counterparts in the modelled world (Domain objects). More objects can be found by using a top-down approach to requirement modelling, whereupon new objects appear throughout the OOA process. *A use case* is an actor's view of the system, where an actor is a user group (user's role). These use cases are dialogues between the users and the system. Use cases are maintained through the whole system development process, and in this way the usability of the system with respect to the users is assured. A use case can be written out as a textual dialogue or scenario in which the required objects and services can be found.

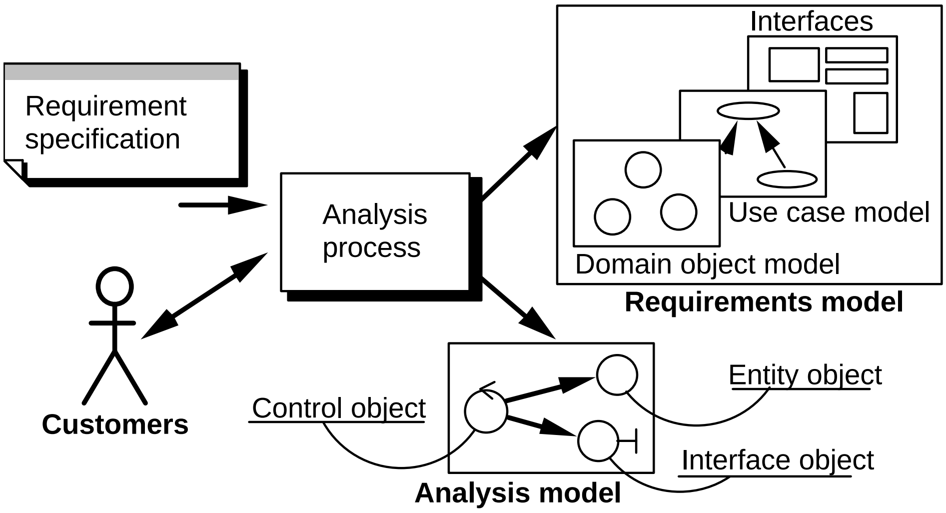

Figure 10. The OOA-process for system modelling [Jacobson et al. 1992]

5. Solution principles for the functions of the product

So far any of the detail structures of the cocoa machine have been discovered. The system has been defined by the user's point of view. By using these introduced OOA-methods the user interfaces and some needed components in the structure have been discovered. We have defined *what* the system should do but we have not defined *how* these properties are going to be implemented in it. We can make for example a virtual prototype of the machine (digital mock-up) and test the functions by using the defined user interfaces. The transformation process has been refined during this modelling phases.
In domain theories by [Andreasen 1998] there is a horizontal causality between different domains and a vertical causality which is the hierarchy in each domain.

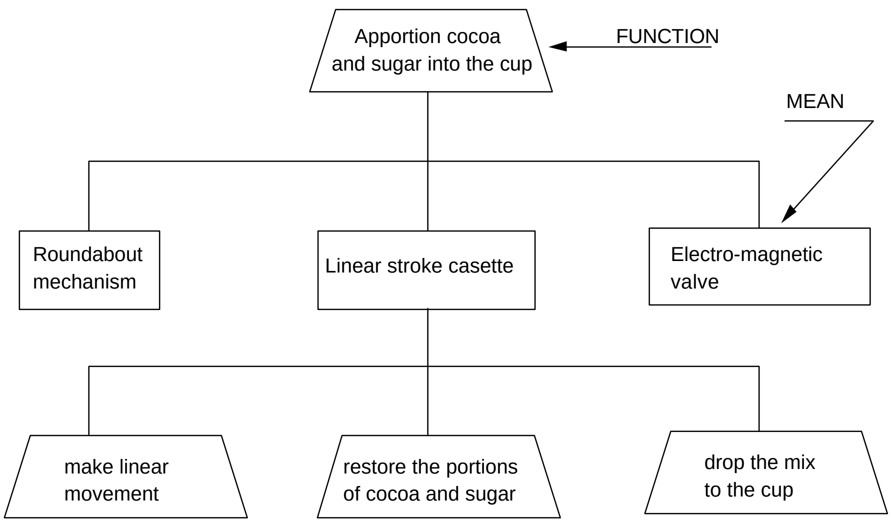

Figure 11. The function/means -tree for the apportion of cocoa and sugar into the cup

The necessary effects (functions) are needed to achieve a transformation in the technical process. The functions are realized by organs, which are the means to achieve the effects. The organs are materialized by machine parts. The cause of an organ is a needed effect (function) and the consequence is a function (or functions) on a lower level. These causalities lead to functions/means-tree, which is used to find out the solutions for functions in the domain theories and it is called also the Hubka's law [Andreasen 1980].

A solution principle of the cocoa maker is presented in the figure 12. In this concept there has been used the roundabout mechanism for the apportion of cocoa and sugar and a four bar lingage for the apportion of milk powder to the cup. The apportion mechanisms are powered by a electric motor

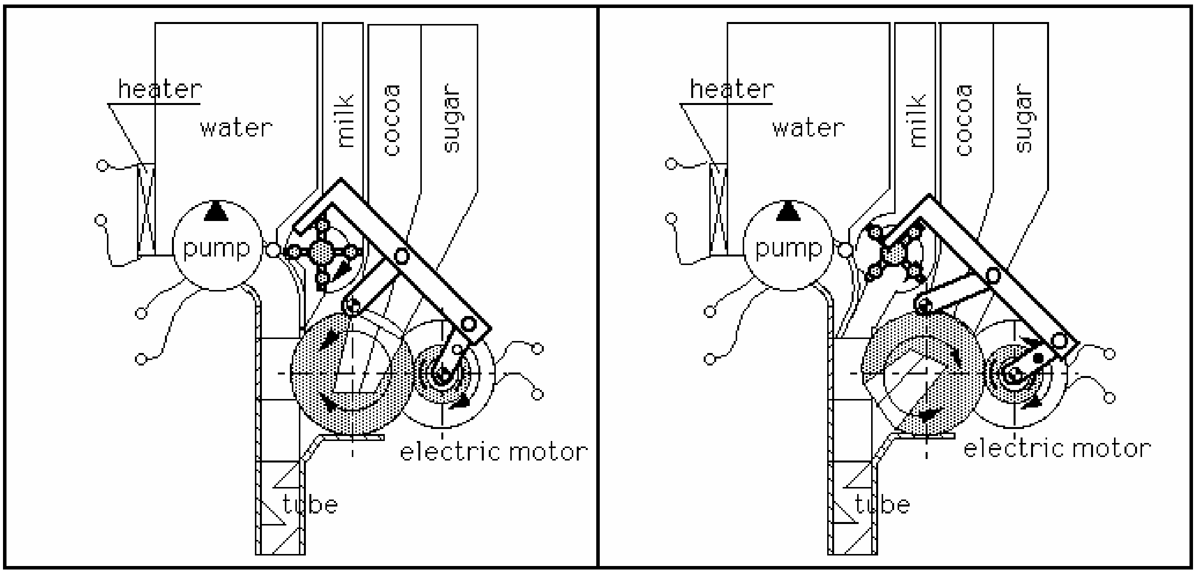

Figure 12. A solution principle of a cocoa maker

6. Conclusions

Mechanical products are complicated, because usually the functions of them are spread all over the structural elements. The functional elements do not necessarily match the building blocks of the structure. The architectures of good mechanical products are more like integral than modular. That is why it is a difficult job to find out the chains from the properties of a product to the principal solution and structural features behind them. Design methodologies, introduced in the literature of design science, help the designer to concentrate on essential matters, when they start to analyze referring or competitive products for product development processes.

Object-oriented methods offer new possibilities to capture the requirements of a mechatronic products. Modelling the user interfaces and making scenarios for typical users (use cases) of the system helps to refine the needed transformation processes. In spite of information modelling for the embedded software in the system (OOA => OOD => OOP), object-oriented analysis helps the designer to find the necessary transformations, which have mechanical or electrical solution in the final technical system.

The third generation object-oriented process modelling language: "Unified Modeling Language" (UML) [Booch et al. 1999] offers formal methods for this definition process in carrying out using actors, use cases, classes, and sequence and collaboration diagrams, which is called the Use Case development in the reference article: "Use of UML/CS SI Development Process" written by August Canals [Canals 2001]. These implementation free models of the system are refined and updated all the time during the whole product development process. Besides defining the needed information and behaviour of the system modules, these models capture and document the real creation story of the product.

References

Andreasen M.,M. (1998) Design Methodology, a PhD course on theory, research and education, Technical University of Denmark, Denmark.

Andreasen, M.,M. (1991) The Theory of the Domains - Workshop on Understanding Function and Function to Form Evolution, Cambridge University, UK..

Andreasen M.,M. (1980) Machine design methods based on a systematic approach - contribution to a design theory (in Danish), Doctoral dissertation, Lund University, Sweden.

Booch, G., Rumbaugh, J., Jacobson, I. (1999) The Unified Modeling Language User Guide, Addison Wesley Longman, Inc., USA, 482 p.

Canals, A. (2001) Use of UML/CS SI Development Process, JOOP, 13(12): 10-17, April 2001.

Hubka V., Eder W.,E. (1988) Theory of Technical Sysytems, A Total Concept Theory for Engineering Design, Springer - Verlag, Berlin, Germany, 275 p.

Jacobson I., Christerson M., Jonsson P. & Overgaard G. (1992) Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, Wokingham, England, 524 p.

Pahl, G., Beitz, W. (1986) Konstruktionslehre: Handbuch für Studium und Praxis. 2. Auflage, Springer-Verlag, Berlin, 590 p.

Shlaer S., Mellor, S.,J. (1992) Object Lifecycles: Modeling the World in State, Yourdon Press, Prentice-Hall Inc., Englewood Cliffs, New Jersey, USA, 251 p.

Tapio Korpela, Lic. Tech.
University of Oulu
Department of Mechanical Engineering
P.O.Box 4200
FIN - 90014 UNIVERSITY OF OULU
FINLAND
Tel: +358 8 553 2052
Fax: +358 8 553 2026
E-mail: Tapio.Korpela@me.oulu.fi