

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Modelling Approach for Redesign of Technical Processes

Ivan Lopez-Arevalo¹, Victor Sosa-Sosa¹ and Saul Lopez-Arevalo²

¹Information Technology Laboratory, Cinvestav-Tamaulipas

²School of Electrical and Electronic Engineering

University of Nottingham, University Park, Nottingham

¹Mexico

²UK

1. Introduction

Nowadays the design and development of new products or modification of existent ones (redesign) is a key and fundamental element to enhance innovation and competitiveness of industrial companies. Design has an increasing importance to differentiate one product from another.

In general, design is the process of specifying a description of a product that satisfies a set of requirements (Umeda 90). Redesign is the process of changing the description of an existent product to satisfy a new set of requirements (Brown 98). Design engineering includes both design and redesign. In the literature we can find diverse terms to narrow design and redesign, such as preliminary, conceptual, functional, creative, routinary, non-routinary, personified, parametric, innovative, etc., but the characteristic activities of the global design engineering can be divided as follows [Subba-Rao 99], see Figure 1:

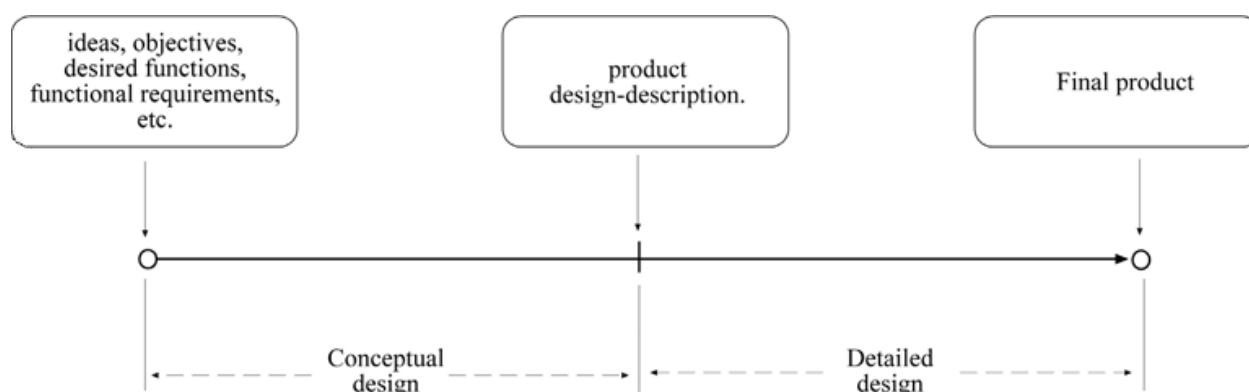


Fig. 1. Product design path.

- Conceptual (re)design, the phase where the global goals, requirements and operation of the product are established based on abstract concepts. The research presented in this chapter deals with this aspect.

- Detailed (re)design, the phase where the results of the conceptual design are used to physically implement a product.

Design engineering involves a wide range of activities. Design engineering can appear in a broad variety of domains, from the assembly of brakes to complex industrial plants and from simple chips to the most advanced super computers. Both design and redesign consist of two main components: the (re)design process and the (re)design object. The (re)design process involves all the (re)design activities performed over the (re)design object, which is the subject entity to be (re)designed. In engineering domains is common to refer to the (re)design object as the artefact. An artefact is a type of product to denote physical and technical devices. The (re)design process over an artefact is characterised by the map between functional requirements to structural requirements. Design and redesign, both can be considered as a dialectic process between goals (what it is desired) and possibilities (real constraints), directed to the satisfaction of functional specifications and performance (Stephanopoulos 90b).

The industry deals with complex technical processes where their behaviour is only predicted by means of complex numerical simulators. The redesign of a process is sometimes necessary when certain time has passed from its implantation or when they must adapt to economical, technological, or environmental requirements. The redesign is not part of the maintenance stage but must be considered into the process life cycle.

From a general point of view, the redesign is done typically in three steps: design-description acquisition (modelling), problem analysis (diagnosis) and proposal of modifications (generation of alternatives). In real redesign situations, human designers intuitively create mental abstract models by removing superfluous information about the process. Such models are based on functions of the components of the process and its context.

From the early 60's, Artificial Intelligence techniques have been used for design, such as constraint-based systems, case-based reasoning, model-based reasoning, planning, neural networks, and genetic algorithms. Although in these approaches the modelling and simulation of the processes has been solved in acceptable way, another problem has been generated, the used knowledge representations require so detailed information that sometimes it is difficult to understand.

The main objective of this work is to obtain a support framework to assist the human designer in the redesign of complex technical processes. The structure of this framework must be based on the common redesign activities performed by human designers on real redesign situations. Therefore, the framework must able to reduce the complexity of the processes to be redesigned, and therefore facilitate the redesign activities.

The framework was obtained integrating model-based reasoning and case-based reasoning techniques. Using model-based reasoning the original process can be modelled hierarchically. Using case-based reasoning alternative process parts can be obtained from other processes, which have to be adapted into the original process. The framework was implemented in the Chemical Engineering domain due to the complexity of the processes involved and the interaction with experts in the area.

The proposed redesign framework has to be able to deal with complex technical processes. In this sense, the type of processes we are referring to follow some assumptions:

1. The complexity of the process must be high.
2. Complex numerical simulators can be used to model the behaviour of the process.
3. The process is already implanted, which means there is a design solution that satisfies the original requirements of such process.
4. Human designers can understand the process intuitively identifying its functional sections.
5. The process can be represented by functional abstract concepts.

The chapter is organised as follows. In section 2 the contextualisation background about redesign and modelling is given. Section 3 describes the proposed modelling approach for redesign. Section 4 shows the experimentation carried out and the obtained results of a first prototype for the proposed approach. Finally, the section 5 presents some conclusions and remarks.

2. Background

2.1 The process of redesign

The research work related to the process of redesign is huge. Following the different research approaches are presented from a general point of view to more specific one. Some of the most interesting definitions of design that we have found in the literature are summarised as:

- Design is formally a search problem in a large space of objects that satisfy multiple constraints (Chandrasekaran 90).
- Design is the task of devising courses of action to change or create better ones (Simon 96).
- Design starts with an intended activity or use (Maher 97b) and uses available knowledge to arrive at a description of an artefact which will produce those results (Gero 90b).

Defining design is difficult because the term refers both to a product (the object to be designed) and a process (the process of design). The reasoning process involved in design allows moving from a functional concept as a starting point to a product solution. Therefore, the design activity can be seen as an activity of synthesis, which is strongly influenced by the skills and mental models of the designer.

Regarding the process of redesign we have identified the following definitions in the literature:

- Redesign is considered as design in which there is a priori knowledge on the general and specialised functions to be performed and on the working principles to be selected [Salomons 95].
- Redesign is an inherent part of most design processes; in which new requirements or new domain knowledge influence the original design process (Brazier 96); but can also be seen as a family of design methods in itself (Pos 97).
- Redesign is part of design, which proposes suitable modifications free for inconvenience of existent artefacts (Kitamura 99).

Independently of the point of view of redesign, three types of redesign can be identified (Dixon 89): parametric redesign, component redesign, and structural redesign. In order to

perform any of these redesign types, it is essential that some form of knowledge is available that allows the adaptation of existing designs. According to Pos (Pos 97) and based on the previous mentioned definitions of design/redesign, it is possible to distinguish two general points of view about the relationship between design and redesign, these are:

1. Viewing the design as total set which contains redesign as a subset. In order to satisfy this relationship, all the components of the design reasoning process should be satisfied for redesign. However redesign as a specialised subset would not be applicable in the same contexts as the more general notion of design. Here, design is viewed as an iterative process that uses intermediate results to get a final design description that fulfills the requirements. The task of redesign on the basis of a design created earlier produces a new temporary design description that is closer to the specification than the former design description.
2. Viewing both design and redesign as independent sets joined by a small common subset. For this relationship to be satisfied there is an expectation that some crossover or overlap will occur, thus only some of the components of design reasoning will be applicable in the redesign context and viceversa. Here, redesign starts with a previously constructed design description and a new set of requirements. The previously constructed design description must now be modified to fulfill the new set of requirements.

Adopting any of the above points of view, basically minimal differences can be distinguished. In both cases the important issue is to bridge the gap between a set of requirements and an existing design description. We can see that design starts from scratch, however, redesign starts with an existing design description, which is modified until it fulfills the current requirements. Both points of view can be captured by a single spectrum of problem-solving methods for redesign.

Several authors (Akin 82, Chandrasekaran 93, Eldonk 96, Brazier 96, Bridge 97, Pos 97) state that the required knowledge for redesign is based on the following two principles:

- Minimise changes in the current design, and
- Maximise existing properties and benefits of the current design.

Many systems that solve redesign problems have been described in literature (Mitchell 83, Howe 86, Fischer 87, Daube 89, Goel 91, Smyth 96, Eldonk 96, Kitamura 99). However when one takes a closer look at the different variants of the redesign task, subtle differences exist that have an impact on how the task can be performed and what kinds of knowledge are involved.

There are a variety of research works referring to design or redesign; from (re)design of abstract (for example, components in software engineering) to physical entities (for example, a reactor in chemical engineering), for a general review see (Brown 97), for some details see (Akin 82, Mitchell 83, Howe 86, Fischer 87, Mostow 89, Goel 91, Bras 92, Stroulia 92a, Chandrasekaran 93, French 93, Brazier 96, Eldonk 96, Pos 97, Price 97, Umeda 97, Gero 98, Culley 99, Culley 99, Kitamura 99, Kraslawski 00, Grossmann 00, Arana 01, Maher 01).

Models in design and redesign are particularly important to guarantee they represent the intentions by which they were created. In general, the models are abstractions of the reality that guarantees communication of ideas by joining concepts, aggregations and relations

(Bridge 97). Akin (Akin 82) outlines that the representational aspects to determine the utility of a model in design are:

- The represented information must be in a level of abstraction suitable for its intention.
- The contents must be in such a way that they are compatible with the expected results according to the mental representations of the designer.
- The model must be consistent with the reality that tries to reflect.

A substantial amount of research has focused on defining models of design (French 85, Tomiyama 87, Treur 89, Brown 89, Chandrasekaran 90, Gero 90a, Takeda 90a, Alberts 92, Vescovi 93, Ohsuga 97, Brown 97). Most of this research highlights that the modelling of the functionality (or properties) of the design object description is an important aspect of the overall design process.

It is possible to represent explicit knowledge in (re)design by means of modelling functions of artefacts. This facilitates the systematisation of the reasoning and some tasks of (re)design. The reasoning based on functions allows abstracting information of the design on the same way as it is made in the reasoning of the initial stages of the design. The process of design of an artefact starts with the conceptual or functional design followed by the basic design and the detailed design (Stephanopoulos 90a). Within these, the functional design plays the central role since it guarantees the quality of the design and the innovation of the product (Umeda 97, Culley 99). The idea of function is fundamental in design since the work of the designers is to design artefacts that must achieve explicit functions (Chandrasekaran 00). Functional modelling is useful to model the object of (re)design, this modelling of objects enhances the formulation of (re)design strategies and the overall (re)design process. Functional modelling "hides" sections of the artefact structure at a lower abstraction level facilitating the manipulation of the artefact description.

Most of the research work on (re)design considers redesign as a knowledge-intensive field; wherein the processes (e.g., tasks) performed, descriptions of sequencing of processes, descriptions of the information within the system, and knowledge employed to perform a task are explicitly modelled most of the times by means of knowledge-based systems. These modelling frameworks try to model the (re)design so the (re)design object as well as the (re)design process are understandable by humans. To do this, human designers use the object specifications to propose a reasonable (re)design approach need to be understood (Leveson 00). Reasoning strategies employed in (re)design are derived or extensions of the commonly named *problem-solving strategies*. Examples of strategies are hypothesis and test (Hempel 66, White 05), pattern recognition (Doyle 62, Kirsch 64, Mitchell 97), skeletal plan refinement (Friedland 85, Tu 89), heuristic classification (Clancey 85), propose and revise (Goel 89), propose critique modify (Chandrasekaran 90), decision tree search (Raiffa 68, Qi 92), means-ends analysis (Newell 63, Rasmussen 86), and reasoning by analogy (Gick 80, Gentner 83).

In the above strategies, the human designer needs to formulate an explicit model of expertise as an integration of two types of models: a domain model and problem solving method. The domain model corresponds to the (re)design object and the problem solving method model corresponds to the (re)design process. Work on domain modelling has only recently attracted the attention of knowledge based system researchers (Stephanopoulos 90a, Schoen 91, Gruber 93, Skuce 93, Sowa 95, Kitamura 98, Fensel 01b, Gomez-Perez 04).

The problem solving method determines how those entities in the model will be used in the actual problem solving process. Domain specific concepts, relationships, and knowledge pertaining to them are captured in the domain model through ontologies (Chittaro 93, Kitamura 99, Fensel 01b, Kuraoka 03).

Independently of models and strategies employed in the (re)design, it is important that such data and knowledge can be recorded in a consistent manner for the future understanding of the (re)design; this constitutes what is called *(re)design rationale*.

2.2 The role of function in the design process

Functions in design play the central role since it guarantees the quality of the design and the innovation of the product (Umeda 97, Culley 99). Function is regarded as what a design object is supposed to do; it is a manageable representation of the overall behaviour of the object (Price 98). Some authors define function as an abstraction of its intended behaviour strongly related to its context (Gero 90a, Goel 92, Stroulia 92a, Chittaro 93, Brown 97, Chandrasekaran 00). Initially the human designers think in functions before they are concerned with specific properties. Functions can exist at different levels of abstraction, depending on the design phase that one is in and the current focus of the design interest. In preliminary design phases, functions usually are independent of working principle, whereas in later design phases, when the functions have been detailed, they become more and more dependent on the working principle that has been selected. In the following, a distinction between three levels or categories of functions is made:

- *General functions*. (Keuneke 91, Lind 94, Kitamura 98, Bo 99) proposed a restricted list of general functions dealing with the transformation of matter, energy and/or information, which are independent of the working principle.
- *Specialised functions or subfunctions*. Act on flows, forces, moments etc., independent of the working principle.
- *Working principle dependent function*. Salomons (Salomons 95) defines it as the realisation of a specialised function (by means of physical phenomena). Several alternative solutions for fulfilling working principle dependent functions can exist without changing the working principle itself.

2.3 The design process

The design process is a complex and not yet well understood cognitive process conducted by humans (Salomons 95). The design process is related to the process of actions and decisions that are taken during design in order to arrive at completed product design. Models of design processes provide a structured description of a process of design. The models differ in their underlying formalisations and have been represented in structures such as:

- blackboard architectures (Ball 92),
- algorithms (Alberts 93b),
- SOAR (cognitive architecture for developing systems with intelligent behaviour) (Steier 91),
- task models or problem solving methods (Brown 89, Brazier 94, Wielinga 97), or

- agent architectures (Dunskus 95, Berker 96, Lander 97).

The following models of the design process can be distinguished:

- Prescriptive models (Salomons 95)
- Descriptive models (Stephanopoulos 90b, Ohsuga 97, Sumi 97)
- Opportunistic design (French 93)
- Decision support problem (Bras 92, Thornton 93, Ullman 91)
- Theorem problem solving process (Takeda 94a)
- Human learning process (Gero 04)
- Multiagent design (Wood 01, DeLoach 04).

2.4 The design object

The design object is the central "actor" that receives the attention during the overall design process. This can be a model of a component, artefact, process or system. Traditionally, the design object was created by technical drafts; but with the advent of computers, the design object has become a computer model that can be shown, modified and deleted easily. Thus, several models of artefacts have been used in design.

Some authors (Rasmussen 86, Douglas 88, Hoover 91, Lind 94, Turton 98, Leveson 00) have observed that abstractions of the design object are important during the design process to manipulate design objects. In this sense, Hoover (Hoover 91) has observed that:

- the design object evolves through abstractions and refinements.
- abstractions and refinements are selected opportunistically and are characterised by the designer focusing on a few aspects of the design object at a time.
- refinements are made within the framework of abstractions.
- conceptual, layout, and detailed stages are not distinct steps in the design process.

Several research works have been developed about (re)design object manipulation. The most relevant approaches in this issue are model-based design and case-based design.

2.4.1 Model-based design

One of the most used approaches in the manipulation of the (re)design object is model-based design which really is a branch of model-based reasoning (MBR) applied to (re)design. Model-based reasoning constitutes a set of techniques applied in several domains and it is used to create models and reasoning about the domain. Mainly the most used technique from MBR has been *compositional modelling* (Falkenhainer 91, Falkenhainer 92, Nayak and Joskowicz 96), which is an approach to construct a model of an artefact (components, devices, processes, systems, etc.) on the basis of a description of the artefact and a query about the composition of the artefact. The modelling of functions (*functional representation*) (Sembugamoorthy 86, Chittaro 98, Chandrasekaran 00) is crucial in compositional modelling. Functional Representation is a top-down approach to describe functions on devices (function), its structure (structure) and its causal processes (behaviour) of the device that culminate with the achievement of the function. Functional modelling reduces drastically the amount of information if simulation is required (Price 98). The approaches of functional modelling can be classified in two groups:

- *State-based representations.* It uses units of function representation, which are abstractions of behaviour states. Behaviour states and hence functions may be associated even with static objects which do not cause any state change.
- *Flow-based representations.* Flow-based representations are based on the concepts of flow and effort. In this approach exists a predefined set of functions, and functions of all existent components are expressed in terms of these primitive functions. This approach is based on the System Theory (Bertalanffy 50) and its derivatives (Abstraction Hierarchy (Rasmussen 86), Qualitative Process Theory (Forbus 84), and Multilevel Flow Modelling (Lind 90, Lind 94)).

2.4.2 Case-based design

Case-based design is a branch of case-based reasoning (CBR). CBR is a general paradigm to solve problems based on the recovery, reuse, revision and retention of specific experiences (cases) (Aamodt 94). CBR has been applied to component-based systems (Maher 97a) which is, however, mostly concerned with the manipulation of design object descriptions. This paradigm is particularly attractive in domains where explicit models do not exist or its understanding is difficult (Kolodner 93). In CBR, similarities between formal methods implemented in computer programs and informal observations from designers are taken into account (Maher 97a). The applications of CBR can be for classification or synthesis tasks. (Re)design problems are within the synthesis tasks.

The direct or analogical use of previous designs or plans of design can reduce and improve the quality of design because take advantages of previous experiences (Maher 95). CBR is viewed as a redesign process for the adaptation of a case where a new artefact (named goal) is designed to achieve certain function, its physical structure can be inferred in analogical way from some physical, chemical or biological object (named source) whose function is similar to the required function.

CBR has been applied to solve problems of real world; there are several works about case-based design for example (Goel 89, Qian 92, Sycara 92, Bhatta 94, Borner 96, de Silva Garza 96, Maher 01, Price 97).

2.5 Redesign approaches

Here the division between research on design and redesign is remarked, only research from the redesign perspective is presented.

2.5.1 Generic approaches in engineering

Goel et. al. [Stroulia 92a, Stroulia 92b, Goel 94b, Goel 97a] presented a control architecture for model-based redesign in the context of case-based redesign. They state that the redesign task is characterised by small differences in the functions desired of and delivered by an existent known design. The redesign is divided in three subtasks: a) generation of modifications to the structure of the old design, b) realisation of the modifications on the structure, and c) evaluation of the new design. Eldonk et. al. [Alberts 93a, Bakker 94, Eldonk 96] presented a redesign approach based on techniques developed in model-based diagnosis. Eldonk et. al. state that redesign activities are diagnosis and respecification. The

objective of this approach is to find the part of the system, which causes the discrepancy between a formal specification of the system to be redesigned and the description of the existing technical system. Kitamura and Mizoguchi [Sasajima 95, Kitamura 99] proposed a redesign approach based on ontologies of functional concepts. They focus on capturing the rationales of design of an artefact and in organising general strategies of redesign. For the first point, they use an ontology of functional concepts that allows to identify functional structures and to represent automatically part of the design rationale. For the second point, they use an ontology of redesign strategies. This approach consists of the following stages: functional understanding, analysis of requirements, proposal of alternative and evaluation.

2.5.2 Mechanical engineering

Aranna et. al. [Fothergill 95, Forster 96, Forster 97b, Arana 00] proposed a redesign environment called DEKLARE, which supports acquisition, representation and reuse of redesign knowledge. It allows the designer to use design techniques to suggest alternative designs that fulfill specific requirements. Gupta et. al. [Das 94] proposed a methodology that automatically provides suggestions of redesign for reducing setup costs for mechanical parts. This approach is based on the interpretation of the design as a collection of mechanical features. The objective is to generate alternative mechanical features by means of geometric changes of the original parts and adding them to the feature set of the original part. Kim [Kim 93] proposed an approach for redesign of assemblies by means of planning techniques. Kim deals with the absence of required design information using the replay and modify principle. He employs a reverse engineering model to infer information about the process executed when creating a given design, and using the inferred information for design recreation or redesign. The propose model consists of the three stages: knowledge acquisition, construction of the default design plan, and redesign based on cases.

2.5.3 Electrical and electronic engineering

Steinberg and Mitchell developed a system to redesign VLSI circuits [Steinberg 85]. This redesign approach is based on planning techniques and causal and teleological reasoning [de Kleer 79]. The subtasks of this approach are: a) focus on an appropriate section of the circuit, b) generate redesign options to the level of proposed specifications for individual modules, c) rank the generated redesign options, d) implement the selected redesign option, and e) detect and repair of side effects resulting from the redesign. Maulik et. al. [Maulik 92] proposed the use of optimisation techniques to redesign CMOS analog circuits. The optimisation approach is guided by three principles. First, equations that describe device characteristics are encapsulated and separated from equations that describe the performance of the circuit topologies. Secondly, constrained optimisation techniques are employed to synthesise the redesigned-scaled CMOS circuit. Finally, constrained optimisation allows the solution of some final constraints over specific variables. Based on the approach of Umeda et. al. [Umeda 92], Tomiyama et. al. [Umeda 94] describe an extension of their approach taking into account the potential functions of the components of an artefact to redesign it. The architecture consists of sensors, which monitor the machine, and a model-based reasoner diagnoses faults and plans repairs. The system generates a FBS model (Function-Behaviour-State) based on the design object, and then searches the model for candidate redundant function. The FBS model consists of a function hierarchy that represents the

designer's intentions, and a behaviour network that describes how the function hierarchy is carried out. The system first tries a control type strategy that adjusts various machine parameters. If the strategy fails the system applies a strategy based on functional redundancy, it uses the potential functions of existing parts in a slightly different way from the original design. Heo et. al. [Heo 98] presented a redesign approach of digital electronic systems by means of evolutive programming. They use directed acyclic graphs known as task flow graphs (TFG) to represent the redesign object. Each node of the graph represents computational tasks; an edge represents a transfer of data. The design process consists of five tiers: a) system-level design, b) architectural design, c) logic design, d) circuit design, and e) physical design.

2.5.4 Chemical engineering

The (re)design of chemical processes is made with the purpose of adapting existing processes to changes in economic, technological or environmental requirements. In the eighties, there were mainly significant advances on saving energy by means of two constraint-based approaches: a) pinch methodology (Tjoe 86, Smith 87, Linnho. 88) and b) mathematical programming on synthesis and design of processes [Papoulias 83, Pistikopoulos 87, Vaselenak 87]. In the nineties, Gundersen [Gundersen 90] made a revision of systematic methods of redesign of processes, which were broadly tackled. In such revision, he emphasised two important observations:

- Most of the projects in the industry of processes were redesign projects.
- The systematic methods of redesign of processes are based on methods of design of processes.

Doherty et. al. [Fischer 87] developed a systematic procedure of redesign by means of opportunistic searches; the procedure considers modifications in the structure of the flowsheet and in the dimension of equipments. Kirkwood et. al. [Kirkwood 88] implemented a methodology of redesign by means of an expert system by using heuristic rules to construct hierarchical structures. Nelson and Douglas [Nelson 90] developed a systematic procedure considering alternative reaction routes; the procedure is hierarchical and provides guides to identify viable processes. Rapoport et. al. [Rapoport 94] proposed an algorithm to design units of process by means of the redesign of already existing ones. The algorithm consists on hierarchical levels and heuristic rules; this approach is similar to synthesis of processes. Stephanopoulos et. al. [Han 95] developed an approach based on agents to synthesis of processes; they modeled the process of design like a set of tasks that can be executed by agents. Systems have also been developed to satisfy economic, environmental and safety constraints. Kraslawski et. al. [Kraslawski 00] developed a methodology centred on the identification and elimination of bottlenecks in reaction and separation sections. Sylvester et. al. [Sylvester 00] optimised processes within the concept of Greener Process. Hertwig et. al. [Hertwig 01] applied techniques of MINLP (Mixed-Integer Non-Linear Programming) to optimise configuration of processes. Pasanen [Pasanen 01] developed a tool for conceptual design of processes, this is called Phenomenon Driven Process Design (PDPD). This methodology focuses on the systematisation of design conceptual of chemical processes. Uerdingen et. al. [Uerdingen 01] presented a screening method based on an analysis of the flow path pattern. They use performance indicators to rate the economic impact of each component of the flowsheet in the flow path.

2.6 Modelling

2.6.1 Modelling the redesign process

The overall redesign process depends on the problem-solving strategy used. In order to start the redesign process, the problem must be specified in terms of objectives that the original artefact must satisfy and the criteria that can be used to rank the alternative designs. Then a synthesis process takes place and the results are a set of alternative designs. Each of these alternatives is analysed and evaluated in terms of the predefined objectives and design criteria. Finally, one alternative is selected to be implemented. The process is highly iterative; the results from later stages are feedback to early stages to modify objectives, criteria, design alternatives, and so on.

Design alternatives are generated through a process of analysis of system composition. The designer breaks down the system (artefact) into a set of subsystems (components), together with the functions and constraints imposed upon the individual subsystem designs. These aspects are analysed with respect to desired system performance features and constraints. The process is iterative until an acceptable design alternative is achieved. At the end of this process all components must be described in such detail that an implementation of the whole object can be performed.

2.6.2 Modelling the redesign object

The understanding of the redesign object depends strongly on the mental models of the human designer. Usually designers communicate their ideas more easily in terms of abstract, high-level descriptions to describe complex concepts (Price 03). The description of the redesign object can be done in many different ways, depending on the context and purpose for which the description is to be used. In the early phases of redesign, highly abstract descriptions (e.g. qualitative or causal) might be helpful, whereas in later phases, more detailed and quantitative descriptions provide more suitable information.

Considering the notion of function, some researchers (Sembugamoorthy 86, Goel 89, Franke 92, Keuneke 91, Chittaro 93, Iwasaki 93) organise the knowledge in a domain by means of functional concepts. The main claim of these approaches is that functions and intentions can provide important additional information for understanding and reasoning about the structure and behaviour of physical systems. In addition, other researchers have directed their extensions to hierarchical modelling by means of different aggregation levels (Liu 91, Rajamoney 91) or different approximations (Weld 86, Kuipers 87, Struss 91, Falkenhainer 91) to organise the knowledge. Independently of the tools and representations employed, several authors (Fischo. 78, Checkland 81, Jaffe 91, Vicente 92) suggest that two important aspects must be addressed if computer tools are used to tackle activities of complex systems:

- *Content*, the semantic information that should be contained in the representation given the goals and tasks of the users. The content gives the basic issues to understand the information about the redesign object. Independently of the amount and complexity of the information, the designer can conceive, in general terms, the objectives of the redesign object.
- *Structure*, how to design the representation to facilitate that the user can extract the required information. The structure concerns to the organisation of the process components.

Rasmussen (Rasmussen 85) observed that the complexity of a system depends on the level of resolution in which the system is considered. The complexity can only be measured comparing with other systems observed at the same level of abstraction. The complexity can be manageable with more or less detail in the representations: then, hierarchical modelling can be seen as a way to handle complex systems. Models of complex artefacts can be expressed in terms of a hierarchy of levels of organisation, each one more complex than the previous. This modelling approach is named *Hierarchy Theory* (Rasmussen 86). Rasmussen studied the protocols developed by people working on complex systems and found that human users structure the system along two dimensions: a part-whole abstraction and a means-ends abstraction.

Some authors (Umeda 90, Franke 92, Lind 94) propose representation approaches for physical systems which maintain a clear separation between knowledge of structure and behaviour on one side and knowledge of function or purposes on the other side. This feature makes them useful in redesign of technical complex systems. The hierarchical functional modelling approaches employed in the present work are Multilevel Flow Modelling (MFM) (Lind 90, Lind 94, Lind 96, Lind 99) and Multimodelling (Ziegler 79, Praehofer 91, Chittaro 93), which are following described. Both approaches provide a more intuitive vision of reasoning on each task to be performed, and thus the redesign activities are enhanced. These approaches have been applied successfully in diagnosis and control domains.

2.6.2.1 Multilevel flow modelling

MFM provides a graphical and systematic basis for using means-end and whole-part hierarchical decompositions in the modelling of complex systems such as industrial plants. By the distinction between means and ends, a system is described in terms of goals, functions and the physical components that involves. At the same time, each of these descriptions can be given on different levels of whole-part decompositions. The main types of decomposition are illustrated in Figure 2. These are functional models with a very high level of abstraction, combined with a teleological representation of goals, or purposes, of the modelled system. Lind has suggested a syntax for a formal language and given the general ideas on how to use the MFM representation.

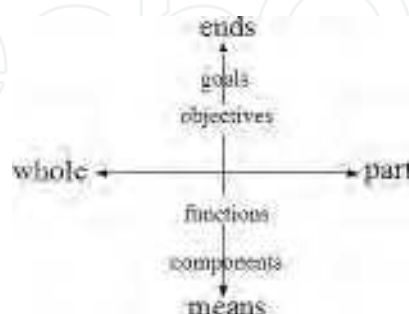


Fig. 2. Means-ends and part-whole dimensions in MFM.

An MFM model is a prescriptive description of a system, a representation of what it has been designed to do, how it should do it, and with which information it should do it. Thus, the three basic concept types of MFM are:

- *goals*, which are the objectives or purposes of the system, i.e., the ends that the designers and operators want that the system reaches.
- *functions*, which are the means by which the goals are obtained, i.e., the powers or capabilities of the system.
- *physical components*, which are the different elements of the system, the equipment of which it consists.

The concept of goal is central to MFM, as it is the "descriptor object" for teleological information. It is important to be able to recognise and describe goals, as they play an important role in every activity using means-end information. Without knowing the goals, it is virtually impossible to know the available functions. Three different types of goals can be recognised:

- *production goals*, which are used to express what enables production. For example, a specific process variable should be kept within a given interval.
- *safety goals*, which are used to express reasons of safe operation. For instance, a particular process variable should be kept above or below some value, or inside or outside an interval.
- *economy goals*, which are used to express considerations of overall process optimisation.

The function is the second important concept on MFM. A function is always associated with a goal, and correspondingly, goals are always associated with functions. MFM describes the functional structure of a system as a set of interrelated flow structures on different abstraction levels. The levels are connected via achievement and condition relations; the flow structures consist of connected flow functions. Thus, the following types of flow structures can be:

- mass flows
- energy flows
- information flows

These flows are of completely different types, but they have many properties in common. Most flow functions can appear in each type of flow structure, thus, there are three flow types of flow functions. In MFM plant functions are represented by a set of mass, energy, activity and information flow structures on several levels of abstraction. The levels are interdependent and form means-end structures. Mass and energy flow structures are used to model the functions of the plant and activity and information flow structures are used to model the functions of the operator and the control systems. The mass and energy flow functions are:

- *source*, the capability of a physical system to act as an infinite reservoir of mass, energy, or information.
- *transport*, the capability of a system to transfer mass, energy, or information from one part of the system to another (from one medium to another).
- *barrier*, the capability of a system to prevent the transfer of mass, energy or information from one part of the system to another (from one medium to another).
- *storage*, the capability of a system to accumulate mass, energy, or information.
- *balance*, the capability of a system to provide a balance between the total rates of incoming and outgoing flows.

- *sink*, represents the capability of a system to act as an infinite drain of mass, energy or information.

These functions can be used to describe information flows. There are also some specific information flow functions:

- *observer*, the capability of a system to translate physical observations to information.
- *decision maker*, represents the decision-making capabilities of a system.
- *actor*, represents the capability of a system to turn information into physical consequences.

In addition to the flow functions, some organisational functions are used. They are concerned with expressing support and control:

- *network*, which is used to group a flow structure and connect it to a goal.
- *manager*, which describes control and supervisory systems, including human operators.

2.6.2.2 Multimodelling

The Multimodelling approach represents many diverse and explicit models of a system, which are used in a cooperative way in specific problem solving tasks. The fundamental assumptions about knowledge modelling and reasoning mechanisms do not identify a unique way of representing a physical system and reasoning about it. On the contrary, the Multimodelling approach is an abstract and general framework that allows for a variety of specific implementations. The fundamental concepts in Multimodelling are:

1. Ontologies. An ontology contains the descriptions of entities in the real system. Two types of ontologies can be distinguished:
 - Object-centred ontology. The real world is made up of individual objects whose properties can be stated in an objective, context independent and general way.
 - System-centred ontology. The real world is made up of systems, intended as organised units, whose elements cannot be defined in isolation.
2. Representational assumptions. This issue concerns about what to represent of the real system in the model. This involves two basic aspects:
 - The scope of the model, i.e., the aspects of the real system which are considered relevant to the purpose of the model.
 - The precision of the model, i.e., the degree of accuracy of the representation
3. Epistemological types. The type of knowledge represented in the model. These types can be:
 - *Structural*. The knowledge about system topology, i.e., the components that constitute the system and how they are linked.
 - *Behavioural*. The knowledge that describes how components work and interact in terms of the physical quantities (variables and parameters).
 - *Functional*. The knowledge about the role components play in the physical processes in which they take part. This knowledge relates the behaviour of the system to its goals, and deals with functional roles, processes, and phenomena.
 - *Teleological*. The knowledge about the goals assigned to the system by its designer and about the operational conditions that allow their achievement through correct operation.

- *Empirical*. The knowledge concerning the explicit representation of the system properties through empirical associations (such as observation, experimentation, and experience). This knowledge may include subjective competence that usually human experts acquire through direct interaction with the system.
4. Aggregation levels. The degree of granularity of the represented knowledge.

3. The redesign framework

The Multilevel Flow Modelling (MFM) and the Multimodelling approaches are able to represent how the human designers behave during the redesign process. MFM is used for high-levels of abstractions and Multimodelling is more suitable for the intermediate and lower levels. Thus, the structure and behaviour of the components (equipments) are abstracted using the Multimodelling approach and then this abstraction is mapped to the MFM approach. The bridge between both approaches is the functions for the equipments in the domain. This functional modelling is the basis to manipulate the process during all the redesign process. The main idea is to model hierarchically the process and reason by using functional abstract concepts. In this way the designer can "navigate" in top-down and bottom-up directions in the representation in similar way as when the designer creates its mental models about the process. From an abstract point of view, there are three actors that play an independent role in this framework:

1. *The simulator*. The commercial software used to obtain the design description of the process and to implement and evaluate the generated alternative process designs.
2. *The reasoner*. The software modules required to model the process, identify the suitable equipment/section to be modified and obtain similar equipments/sections based on the selected equipment/section.
3. *The human designer*. The human user interpreting the results.

The stages of the proposed redesign framework (Figure 3) are:

1. Identification of objectives and criteria. This stage covers the design-description acquisition and the identification of candidates stages of the framework.
2. Generation of design alternatives. This stage is similar to the obtaining alternatives and adaptation stages in the framework.
3. Evaluation of alternatives. This stage is carried out manually by the human designer.
4. Implementation of alternatives. This stage is also carried out manually by the human designer.

This redesign framework can deal with complex technical processes (the redesign object). The modelling approach was chosen to mimic the behaviour of human designers in real redesign situation of such processes. The final intention is to support human designers, not to carry out the redesign automatically without human intervention.

3.1 Design-description acquisition

The first stage of the framework is to obtain the design description of the process to be redesigned. This description is enough to carry out the redesign activities, and just few adaptations are necessities to fulfill the redesign objective. This stage is carried out in two substages: data acquisition and functional identification.

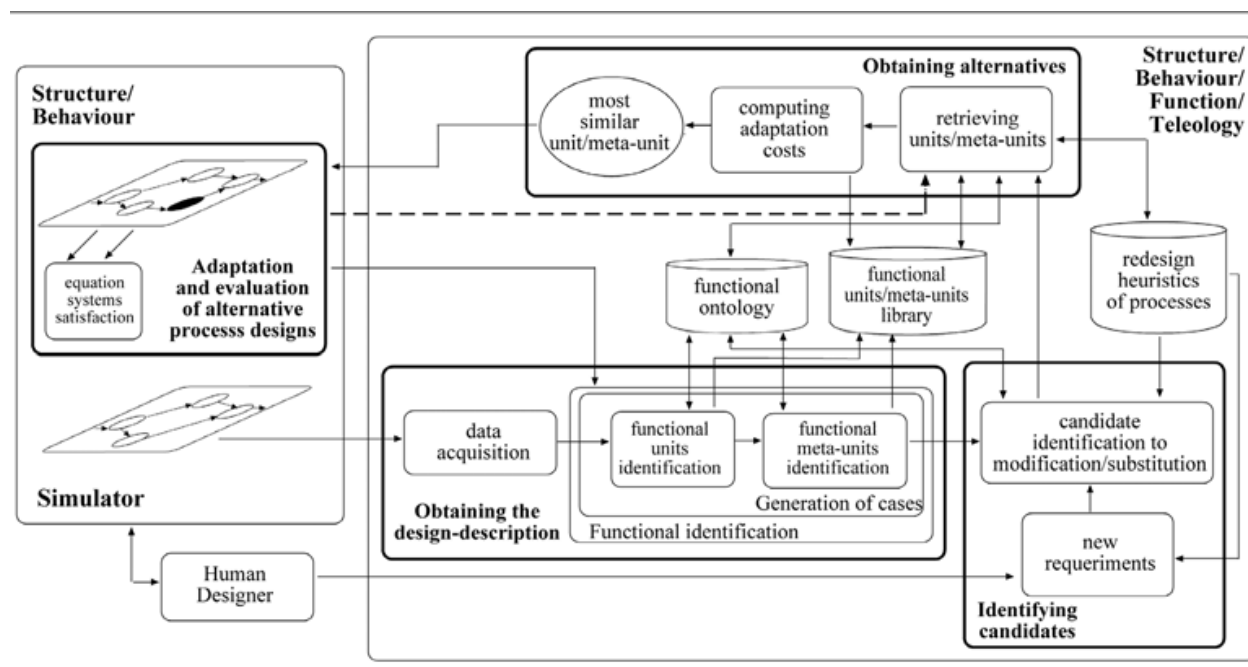


Fig. 3. The proposed redesign framework.

3.1.1 Data acquisition

It deals with data extracted from the specialised simulator used to implement the process to be redesigned. It was conceived as an appropriate step to reduce human intervention on the introduction of data to the reasoner module. The aim of the data acquisition is to obtain only the most useful data to generate the appropriate knowledge useful for the redesign of the process; thus irrelevant or superfluous data is ignored. Based on this data, the following types of knowledge are generated:

- *Structural*. Knowledge related to the topology of the process, i.e., the equipments conforming the process and the connection between them.
- *Behavioural*. Knowledge related to the values of variables and parameters that characterise the behaviour of each equipment.

3.1.2 Functional identification

The data obtained from the simulator is used to model hierarchically the process. To do this, the functions of each equipment in the process must be identified. Based on the identified functions, the functional sections of the process can incrementally be identified. In the rest of the chapter any equipment will be named *unit* and a functional section will be named *meta-unit*. This stage is divided into functional unit identification and functional meta-units identification. Here it is necessary to specify an ontology about the functional issues of the existing equipments in the process. By using this ontology, it is also required to specify a priority order of functions and the process variables related to each one. The grouping of functions depends strongly on such priority order.

Functional unit identification

The function of each unit is inferred by analysing their inputs (preconditions) and their outputs (postconditions), the variables involved, and the neighbour units. This process involves the analysis of the behaviour of the unit and its consequences in the surrounding units (the units connected to it). The next classification of functions was obtained based on MFM and Multimodelling:

- *Broad function*. Denotes a process-independent function that can be achieved considering only flows of mass, energy, or information.
- *General function*. Denotes a function that can be achieved by several equipments in a domain. These functions deal with the transformation of mass and energy and are independent of the physical phenomena.
- *Specific function*. Denotes the abstract function as it is known in the domain of the process. These functions relate flow variables with a specific physical process. According to the domain, these are functions to denote functional sections into the process.
- *Working function*. Denotes a function that can be achieved by a specific single unit. These functions relate specific flow variables with a specific physical phenomenon.

A unit can have several functions but only one goal (the objective, intention or purpose of the artefact). Several units can have a common goal. Therefore, based on the data extracted from the simulator it is possible to infer the following knowledge:

- *Functional*. Knowledge about the roles of each unit. The functional knowledge connects the behaviour (physical phenomena and processes) of the unit to its goal.
- *Teleological*. Knowledge related to the goals of each unit by considering the required input operational conditions and the output operational conditions that were meant to be produced.

The functional knowledge is independent of the process (the same functional knowledge can be found in others processes), while the teleological knowledge are the goals assigned to the units by the designer).

As result of this stage, each unit is represented by structural, behavioural, functional, and teleological models. The aim of this stage is to model the process in a higher level of abstraction (respect to the simulator).

Functional meta-unit identification

Based on the functions inferred in the functional unit identification stage, it is possible to identify the functional sections of the process (named *meta-units*). The incremental identification of these functional sections denotes the most important sections of the process. This incremental identification is carried out by generating different representations of the process at different levels of abstraction. The function of a unit is a working function because the unit (representing real equipment) was designed to perform only such function. The functional sections of the process denote specific and general functions by means of meta-units. Meta-units representing general functions are composed by meta-units with specific functions, not necessarily of the same type. A meta-unit represents a functional section at an abstract level. Thus a meta-unit at a higher abstract level can contain several

units and other meta-units. Two or more meta-units can generate a more abstract meta-unit. Units/meta-units with lower priority functions are “absorbed” by units/meta-units with higher priority functions, as shows Figure 4.

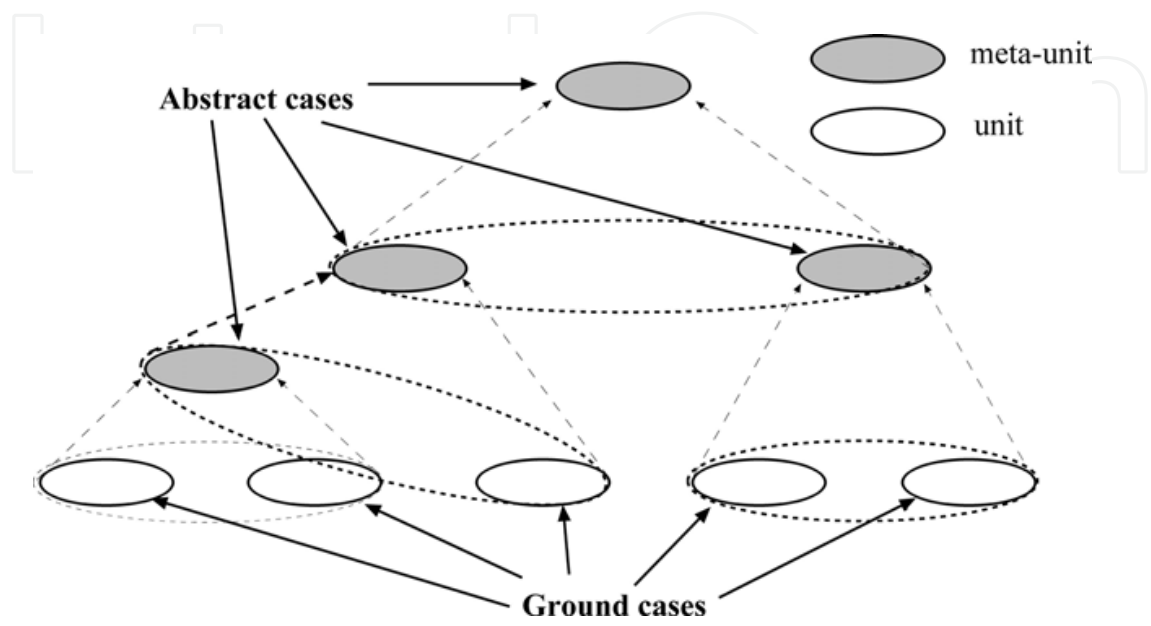


Fig. 4. Grouping of units/meta-units.

Every functional section forms a hierarchy of meta-units and units where meta-models are connected in a same level (intralevel), as shows Figure 5, and at different levels (interlevel), as shown in Figure 6.

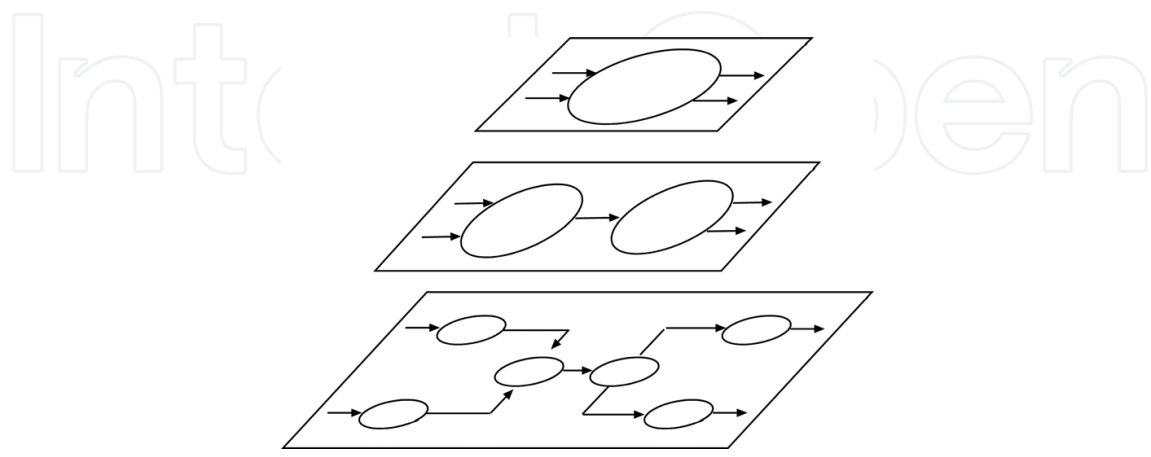


Fig. 5. Intralevel meta-models.

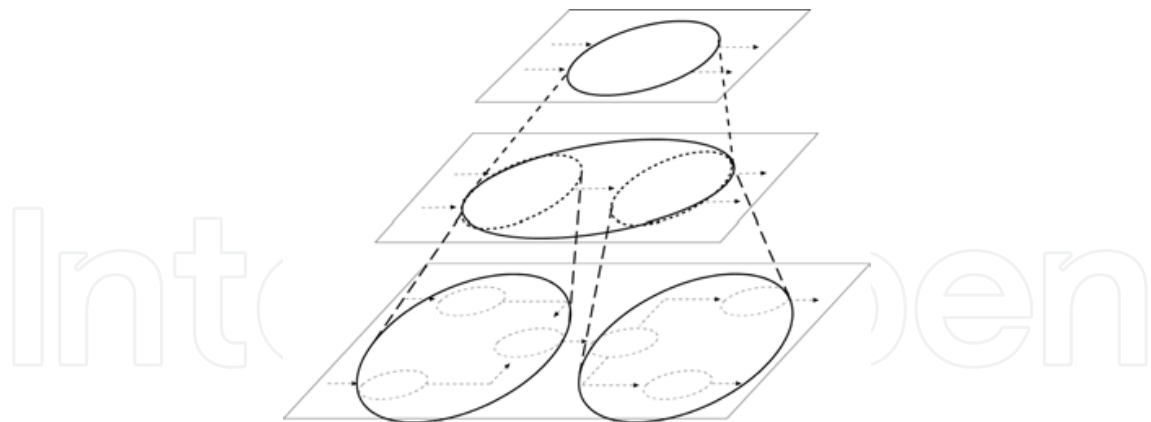


Fig. 6. Interlevel meta-models.

Then, the overall process is represented by several functional sections denoting flow structures. Incrementally the identification of such functional sections denotes the most important sections of the process. This process finishes until the most important functional sections are identified. This corresponds to the "blackbox" from which the original design could begin (Figure 7).

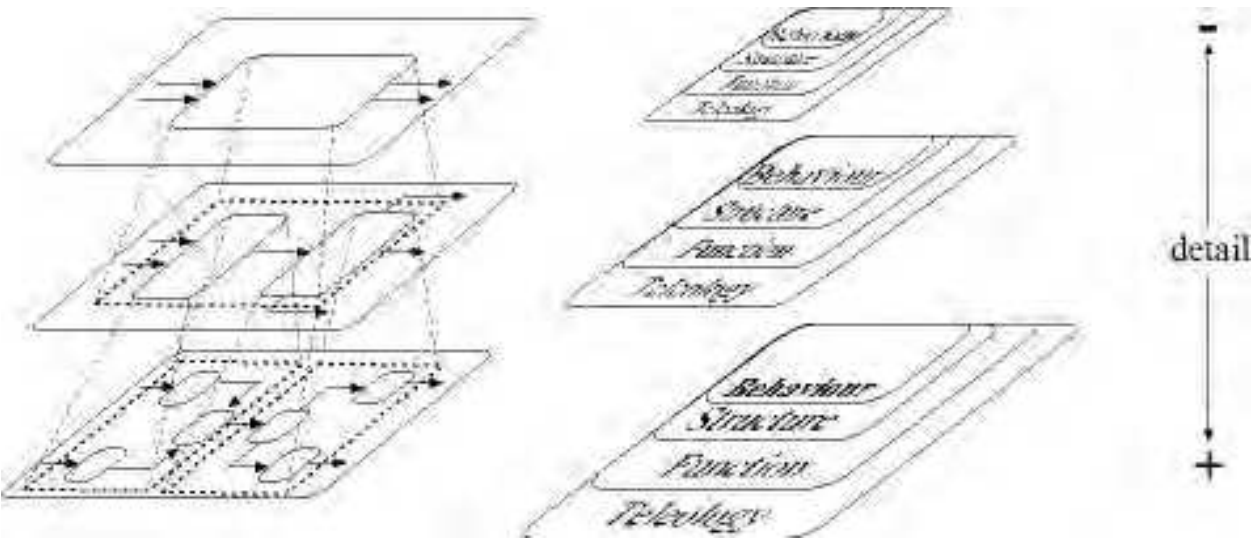


Fig. 7. Abstraction of a process.

3.2 Candidate identification

The aim of this stage is to get the suitable unit or meta-unit to be modified to fulfill the redesign objectives. In a first instance, the redesign must be focused on a process variable. Once the variable is identified, a diagnostic algorithm is used to identify the units/meta-units affecting such process variable. This reasoning process is based on the functions identified at the functional analysis stage. This stage is composed of two substages: specification of redesign requirements and identification of the suitable unit/meta-unit for modification or substitution.

3.2.1 Specification of redesign requirements

Here the human designer must specify the new requirements that the process must satisfy. Two categories of redesign requirements can be identified: functional requirements and physical requirements. A design specification always contains a single functional requirement; it may also contain a set of physical requirements. A functional requirement represents an abstraction of the intended behaviour of the artefact. It can be a general, specific, or working function. There is no direct association between the function that has to be provided and the physical mechanism that provides it. A physical requirement represents an abstraction of the physical process variables, which satisfy the functional requirement specified in the design specification. It denotes preferences about the designer intentions regarding some aspect of the process. The redesign specification can be represented by the functional and physical requirements or only by physical requirements. A redesign specification is a mean (goal) which is defined in terms of functions that must be embodied in a process in order to provide some higher level functionality.

3.2.2 Identification of candidates

Here In this stage, a diagnostic algorithm is used, that works on the functional concepts identified in the functional analysis. Diagnosis helps to detect "faulty" components (those that do not satisfy the global performance of the process). The design description and the new specifications are used to identify the possible candidates for modification or substitution. The diagnostic algorithm returns an ordered list of units or meta-units. Because the diagnostic algorithm operates over abstract functional concepts, no simulation is required. The diagnostic algorithm does not return the exact unit or meta-unit responsible for the "faulty" behaviour, it returns a list of units/meta-units that do not fulfill the global performance of the process represented by the redesign specification. The human designer is responsible to choose, from the resulting list, the appropriate unit/meta-unit that has to be modified or substituted into the process. Since this unit/meta-unit is connected to others by a flow path, the "cause" and "consequence" units/meta-units also must be identified. A cause unit/meta-unit is(are) the unit(s)/meta-unit(s) situated before the current unit/meta-unit in the flow path. They are responsible to provide the appropriate operational conditions to the involved process variables in the function of the unit/meta-unit of interest. A consequence unit/meta-unit is(are) the unit(s)/meta-unit(s) situated after of the current unit/meta-unit in the flow path. They are the unit(s)/meta-unit(s) affected by the operational conditions given by the unit/meta-unit of interest. Both, the cause and the consequence units/meta-units, are not necessarily the closer neighbours.

3.3 Generation of alternatives

The aim of this stage is to obtain similar units (equipments) or meta-units (sections) to adapt them into the current process based on the suitable unit/meta-unit identified by the human designer at the last stage. The best way to obtain similar units/meta-units is from similar processes. With the adaptation of any retrieved unit/meta-unit into the process of interest, then the alternative process design is obtained, which is the final goal of the redesign framework. An appropriate approach to perform this stage is case-based reasoning (CBR) for reusing past experiences on new situations.

Starting with the selected unit/meta-unit at the candidate identification stage, similar units/meta-units can be retrieved from other processes. The retrieved unit/meta-unit, which functional and teleological models are the most approximate to the functional and teleological models of the unit/meta-unit of interest, is adapted. This process requires that the performance and operational conditions of the cause and consequence units/meta-units associated with the retrieved unit must be similar to the original case.

According to the later stage, the overall target process is modelled as a graph denoting a hierarchy of functions. Therefore hierarchical case-based reasoning (Smyth 01) is required. Thus, based on the levels of abstraction, two kinds of cases are distinguished:

- *ground cases*. Cases located at the lowest level of abstraction, units (real equipments)
- *abstract cases*. Cases represented at higher levels of abstraction, meta-units (non-existent "meta-equipments")

The organisation of cases into the library is performed according to the type of functions of the unit/meta-unit. In this way, several groups can be distinguished according to the general function type: *source*, *transport*, *barrier*, *storage*, *balance*, and *sink*. Within each functional group, units and meta-units are grouped based on their specific functions. Again, within specific functions groups, the units/meta-units are grouped based on the working function achieved. There are not distinctions between units and meta-units with the same specific function. The case library is organised by an abstract hierarchy based on function groups. This structure denotes the organisation of the functional ontology used in the framework.

To retrieve cases from the case library, a similarity engine is used. Only units/meta-units of the same specific functional group are considered. The similarity engine uses functional and teleological targets to search into the library of cases. Functional and teleological models denote strongly the relationship between the units/meta-units and its neighbours. Two types of similarity are computed, local and global, which are defined as follows:

- *Local similarity*. Similarity between two cases is based on the local similarity between each feature of such cases. The computation depends on the type of the feature and the value that it could take.
- *Global similarity*. Once a set of local similarities has been computed for each known feature-value pair, the CBR system computes the global similarity of the candidate cases based on such set.

As final result, a set of cases is obtained which contains meta-units (with its corresponding units/meta-units) or units. The set is ranked according to the global similarity between the target case (the unit/meta-unit of interest) and the source cases (the retrieved units/meta-units).

3.4 Adaptation and evaluation

For this stage the human designer intervention is required since the use of the specialised simulator employed in the data acquisition is mandatory to test the proposed candidates. The adaptation is highly domain-dependent and requires online simulation of the process to verify its correct performance. Since information of abstract cases can not be used directly,

the adaptation and revision of equipment must use information of ground cases (real equipments on the simulator). The human designer must fit the ground cases with the process variables involved.

To facilitate the adaptation, an adaptation cost is computed to suggest the human designer the adaptability of the chosen unit/meta-unit. The adaptation cost is based on the differences of the selected unit (source case) and the cause and consequence units/meta-units identified with the diagnostic algorithm. Thus, the cost is a normalised numerical value denoting the difference on the values of the process variables involved in the performance of the unit and the values of the process variables involved in the performance of the neighbour units/meta-units.

The adaptation cost has a value between 0 and 1. Values close to zero mean the adaptation is difficult. The designer experience is determinant because modifications on equipments may affect the overall performance of the process. The modification of the original process, based on the adaptation of a retrieved case, generates an alternative of the process for every unit/meta-unit adapted.

4. Evaluation and results

The framework has been applied to the Chemical Engineering because chemical processes are suitable for the proposed approach. A chemical plant can be constituted of one or more chemical processes.

The software modules of the redesign framework have been implemented in Java (Sun 05). Additional libraries have been used such as JESS (JESS 08), Ozone (Ozone 08), and The Selection Engine (Wetzel 00). The interaction with the user is done through a graphical interface.

The implementation of the concepts of the proposed redesign framework is based on the following ontological commitments.

- The chemical processes typically operate at steady-state. That means that values of variables do not change with respect to time.
- A chemical process is constituted of real and abstract units. The abstract units are the sections of the process that appear as atomic elements in conceptual models. All real equipments can be viewed as descendants of the generic real equipment.
- A generic real equipment can be modelled as an object having four attributes: structure, behaviour, function, and teleology to describe all the properties of any real equipment.

Since the framework requires functional concepts, a crucial point is to define the type of functions by means of an ontology. The functional ontology obtained is formed by high-level and low-level concepts in a similar way to the SUMO (Suggested Upper Merged Ontology) ontology structure (Niles 01). SUMO structures the concepts using meta-concepts, where terminology of general purpose is situated at higher levels, while terminology to specific domains is situated at lower levels. The ontology developed has extended generic concepts of SUMO such as process, objects and mereological and topological concepts. Additional specific concepts have been defined: physico-chemical processes, thermodynamic processes, substances (mass and energy), substance roles (of

chemical compounds), functional roles, devices (equipments and connections), measure units, tasks, operations, and relations.

Most of the concepts in the ontology correspond to physical entities. The high-level concepts denote very abstract concepts, which can be found in several domains. The middle-level includes the functional concepts proposed in the Multilevel Flow Modelling and Multimodelling approaches, which are: *source, transport, barrier, storage, balance, and sink*. The low-level functional concepts come from the well-known chemical process design methodologies developed by Douglas (Douglas 88) and Turton (Turton 98). The low-level functional concepts can be grouped as: *reaction, separation, temperature change, pressure change, and flow change*. These concepts are called *general functions*. Each specific function is divided into more specific ones named *specific functions*, which denote the function of the equipment into the process. Also each specific function is divided in more specific ones, called *working functions*. A working function can be associated with one or more units and a unit can be related to more than one function. But from the several working functions, only one is the main function in the process.

Over the identified functions an importance functional order and the variables involved in such functions (carried out by the equipments of the process) have been defined. This order was defined with the aim of forming groups of functions where more important functions "absorb" functions with minor importance, see Figure 8.

The framework was tested over 50 chemical processes (Lopez-Arevalo 05):

1. Acetaldehyde from ethanol
2. Acetaldehyde from ethylene and oxygen
3. Ethyl acetate
4. Vinyl acetate
5. Acetone
6. Acetic acid
7. Acrylic acid
8. Cyanhydric acid
9. Nitric acid
10. Acrolein
11. Ammonia from natural gas and pure N₂
12. Ammonia from pure N₂ and H₂
13. Phthalic anhydride from naphthalene
14. Phthalic anhydride from o-Xylene
15. Maleic anhydride
16. Benzene and methane
17. Benzene and o-Xylene
18. Benzene, Toluene and Styrene
19. Separation of Chlorine-Benzene and Benzene
20. Ethyl-Benzene
21. Cumene
22. 1,3-Butadiene
23. Cyclohexane
24. Allyl Chloride

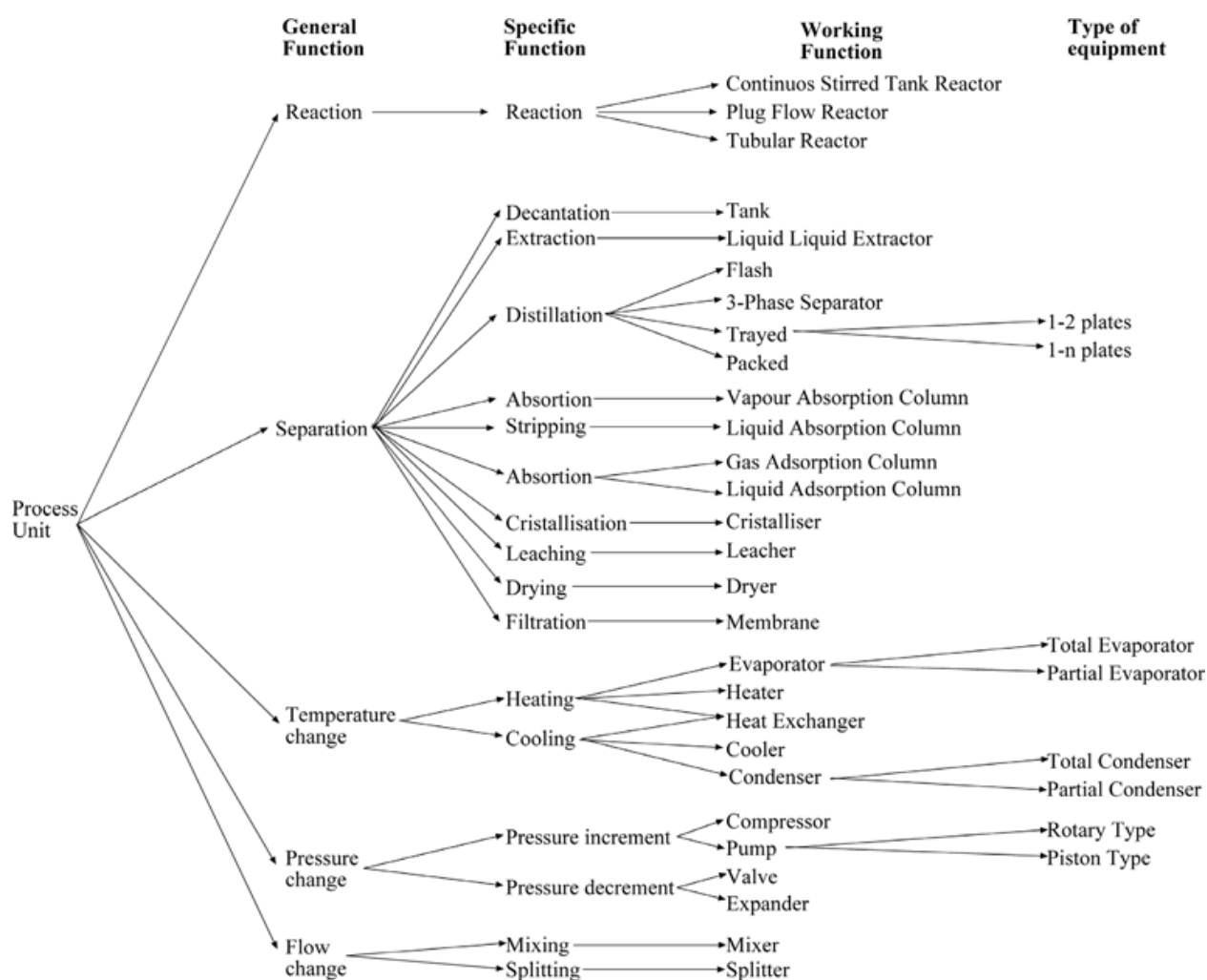


Fig. 8. The hierarchy of functions

25. Separation of Ciclohexane
26. Chloroform
27. Ethanol
28. Purification of Ethanol
29. Dimethyl ether
30. Ethyl tert-butylic ether (ETBE)
31. Methyl tert-butylic ether (MTBE)
32. Tert-amyl Methyl ether (TAME)
33. Styrene
34. Separation of ethane, n-heptane y n-octane
35. Ethylene
36. Ethylene oxide
37. Formaldehyde
38. Formaline
39. Methyl formate
40. HP gas
41. Heptane

- 42. Hydrogen
- 43. Separation of methane
- 44. Separation of methane and ethane
- 45. Methanol from natural gas
- 46. Methanol from carbon monoxide
- 47. Oxygen and nitrogen
- 48. Purification of paraffins
- 49. Propyleneglycol and dipropylene glycol
- 50. Vinyl chloride

4.1 Modelling the ammonia process

The ammonia production process (Figure 9) has been selected as case study because it is one of the most relevant chemical processes in the industry. Ammonia is one of the most important chemicals commodities because of its role in the production of fertiliser and hence of food. It is produced in over 80 countries worldwide with a volume of 130 million tonnes annually (GIA 04).

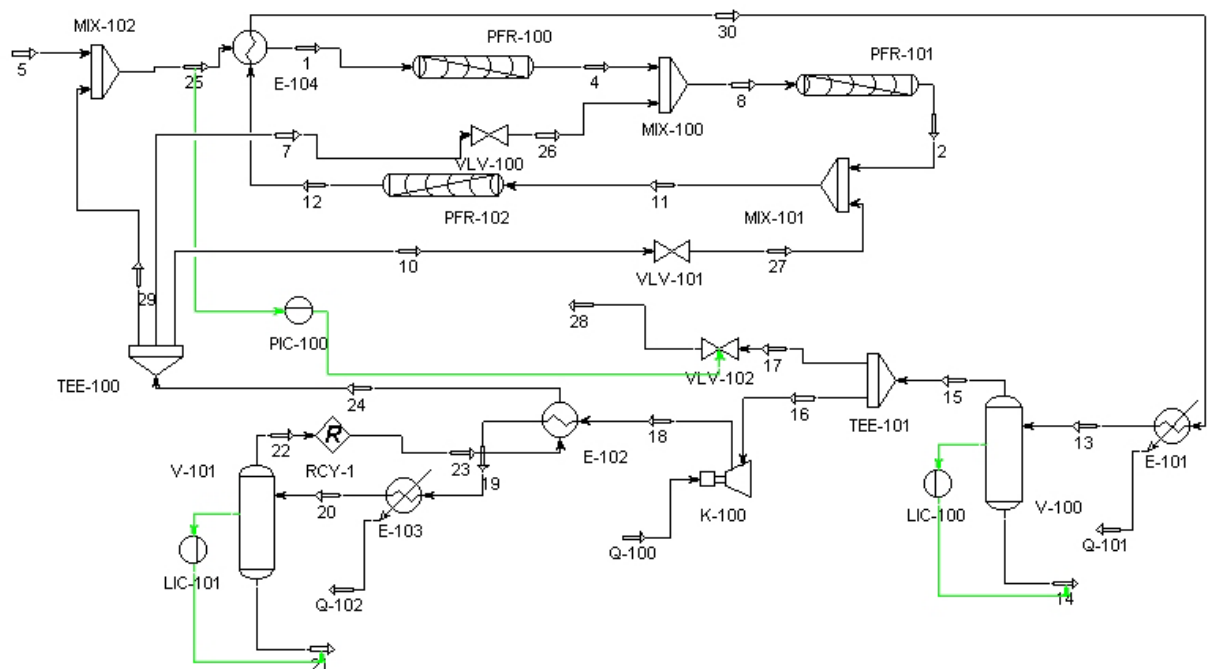


Fig. 9. The ammonia production process

Technical changes on process equipments were taken into account and some other issues such as economical costs, changes in pipes, environment impact, etc. were not considered. To illustrate the performance only the ammonia production process is used as case study.

First, the data is extracted from the Hysys simulator (Hysys 04), and then the "roles" of chemical substances are asked to the human designer. The first level of the process (abstraction level 0) is shown in Figure 10. This GUI allows the user to interact in two ways,

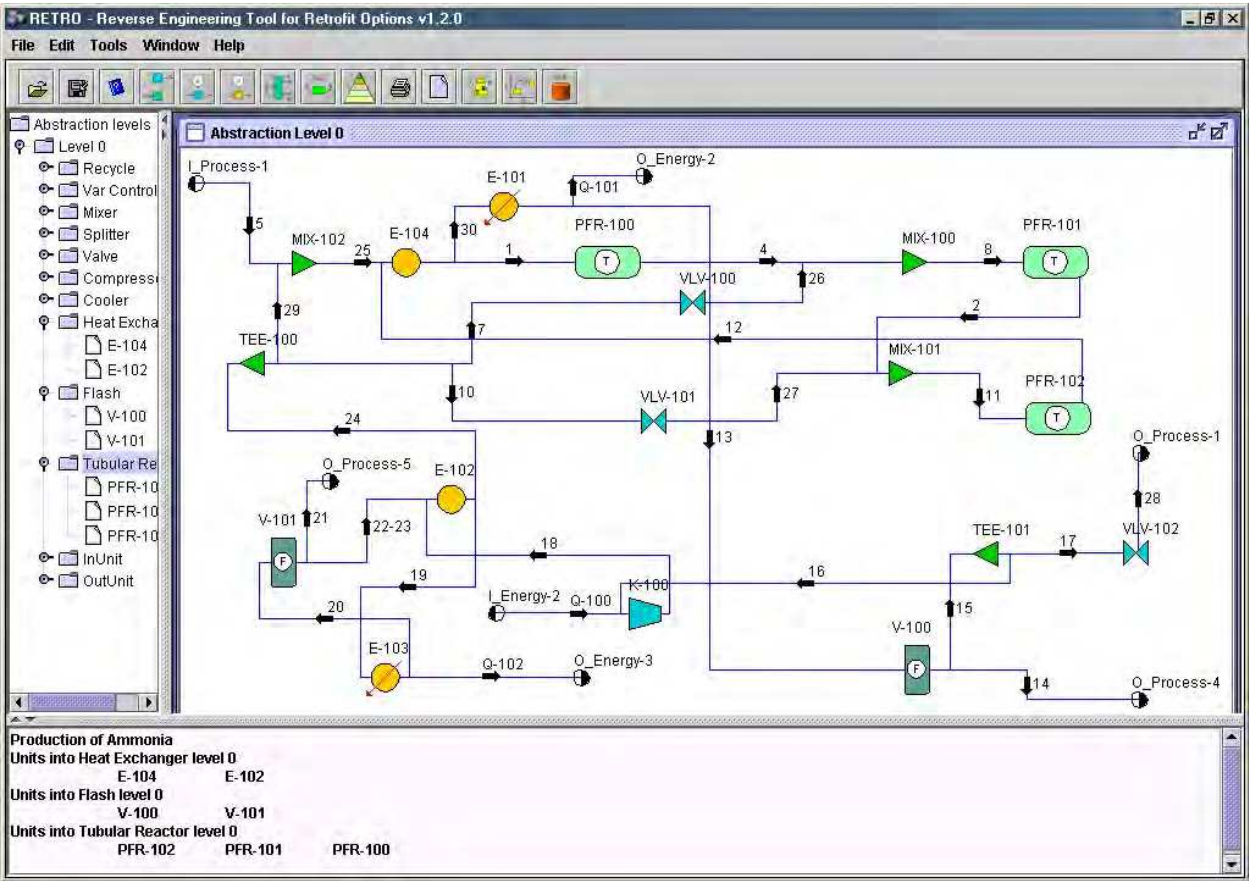


Fig. 10. First representation of the ammonia production process.

through menus and panels. The diagram panel (upper-right panel) allows the user to manipulate the process layout; the user can organise the components of the process according to its needs. The navigation panel (upper-left panel) is used to navigate into the levels of the process; abstraction levels and its corresponding components. The information panel (bottom panel) displays information about operations carried out in the prototypes. Table 1 summarises the type of equipments and its corresponding functions.

For the generation of meta-units, the flow change equipments (mixers and splitters) are grouped to other ones in the abstraction level 1, as it is shown in Figure 11.

The different abstract levels were automatically generated until the final abstract models were created at level 6 (see Figure 12). The abstraction process continues until the whole process is represented by just one meta-unit. Only new units and meta-units are represented to illustrate how the functional groups are created. In this sense, connections between units and meta-units in the same level have not been represented. The scheme represents the process by means of groups of the general class of type of equipment; its general function can be deduced from them. Figure 12 represents all the unit and meta-units in each level, in similar way that they are presented to the designer.

General Function	Specific Function	Working Function	Label
Flow change	Flow increment	Mixer	MIX-100
			MIX-101
			MIX-102
	Flow decrement	Splitter	TEE-100
			TEE-101
Pressure change	Pressure increment	Compressor	K-100
	Pressure decrement	Valve	VLV-100
			VLV-101
			VLV-10
Temperature change	Temperature increment	Cooler	E-101
	Temperature exchange	Heat Exchanger	E-103
			E-102
			E-104
Separation	Distillation	Flash Separator	V-100
			V-101
Reaction	Reaction	Tubular Reactor	PFR-100
			PFR-101
			PFR-102

Table 1. Equipment and functions in the ammonia production process.

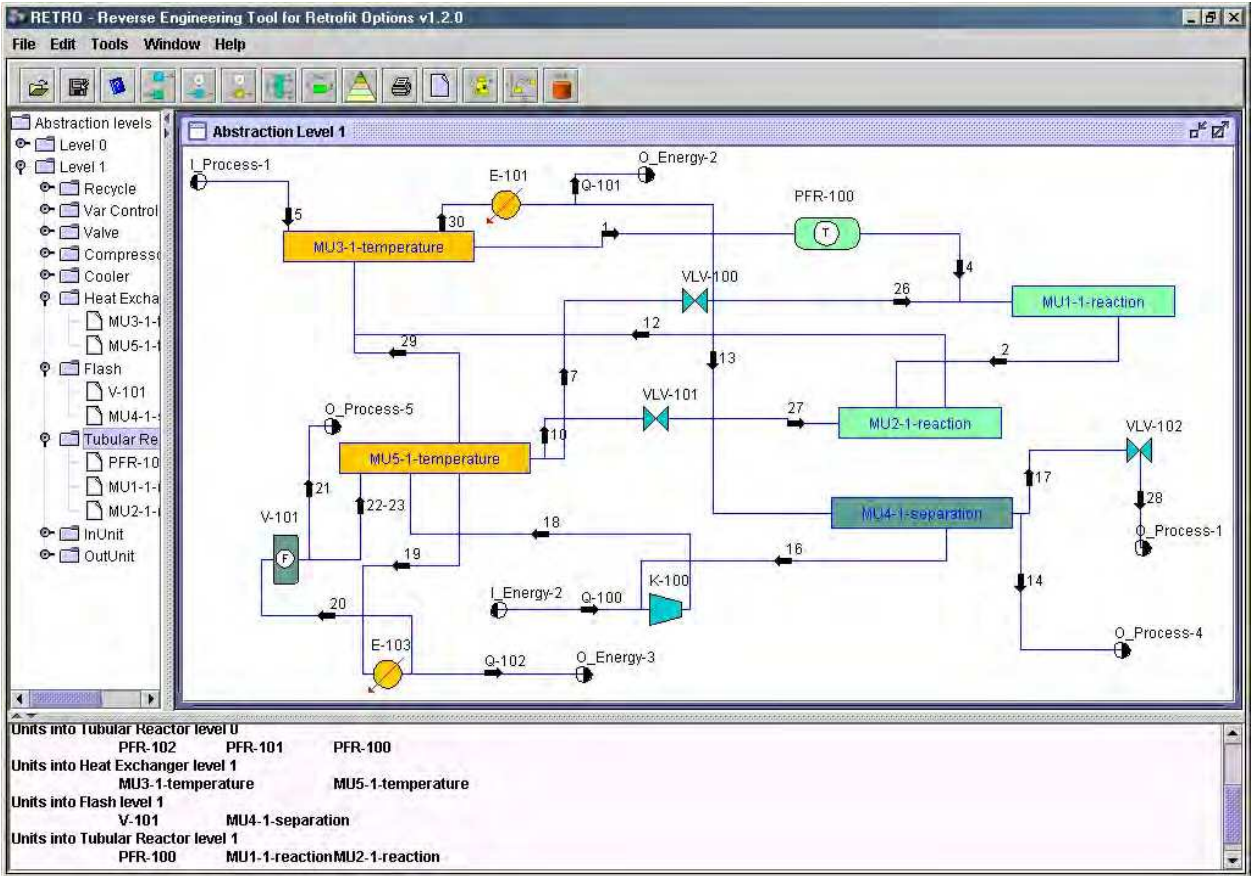


Fig. 11. Grouping of flow change units.

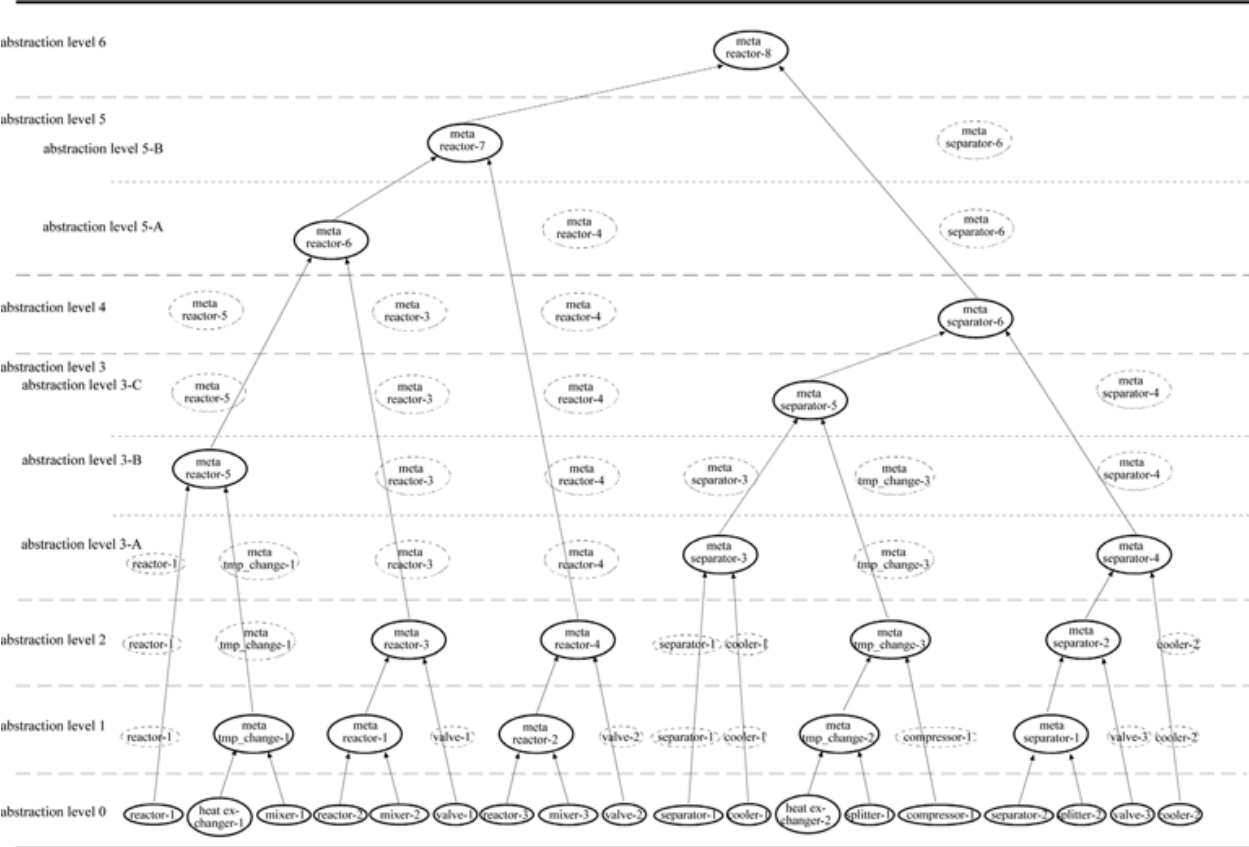


Fig. 12. Hierarchical representation of the ammonia production process in bottom-up direction.

4.2 Identification of candidates

According to the new design objective(s) the process must fulfill, modifications on reactors are used to illustrate the results of the framework:

Requirement

The redesign problem is the increase of the production of ammonia by 15% in the plant represented by the scheme of Figure 10.

The human designer first needs to identify the variables that may affect directly the production of ammonia. Then the designer identifies that the increase of production can be achieved by modifying any of the next conditions:

- pressure
- temperature
- concentration

This gives an idea on the types of equipment the diagnosis must focus on. Assuming that the concentration variable is selected, this is affected by reactors and separators. Initially, reactors affect the concentration of product because they produce the main product, and separators affect it in secondary manner by incrementing the purity of the product.

Therefore, the focus will be on reactors, where the ammonia is originated. Since all the roles of the chemical substances are known, the diagnostic module focuses on the concentration of ammonia (which has the main product role). Thus all values related to this substance are analysed.

We are interested in finding where the main product is produced. The search starts at the highest level in the hierarchy -abstraction level 6- following the flow direction, from left to right. As result of the search, all the units and meta-units affecting the concentration variable have been identified, as shows Table 2.

Abstraction level	Identified components
6	meta-reactor-8
5	meta-reactor-7
4	meta-reactor-6, meta-reactor-4
3	meta-reactor-5, meta-reactor-3, meta-reactor-2
2	reactor-1, meta-reactor-1, reactor-3
1	reactor-2

Table 2. Identified candidates

Since modifications to the process can be performed only at ground level, the cause and consequence units are searched in this level. To illustrate the cause and consequence identification, assume that we focus on the meta-reactor-3, which at ground level includes the units PFR-101, MIX-100, and VLV-100 (Figure 13). In the state analysis, the state

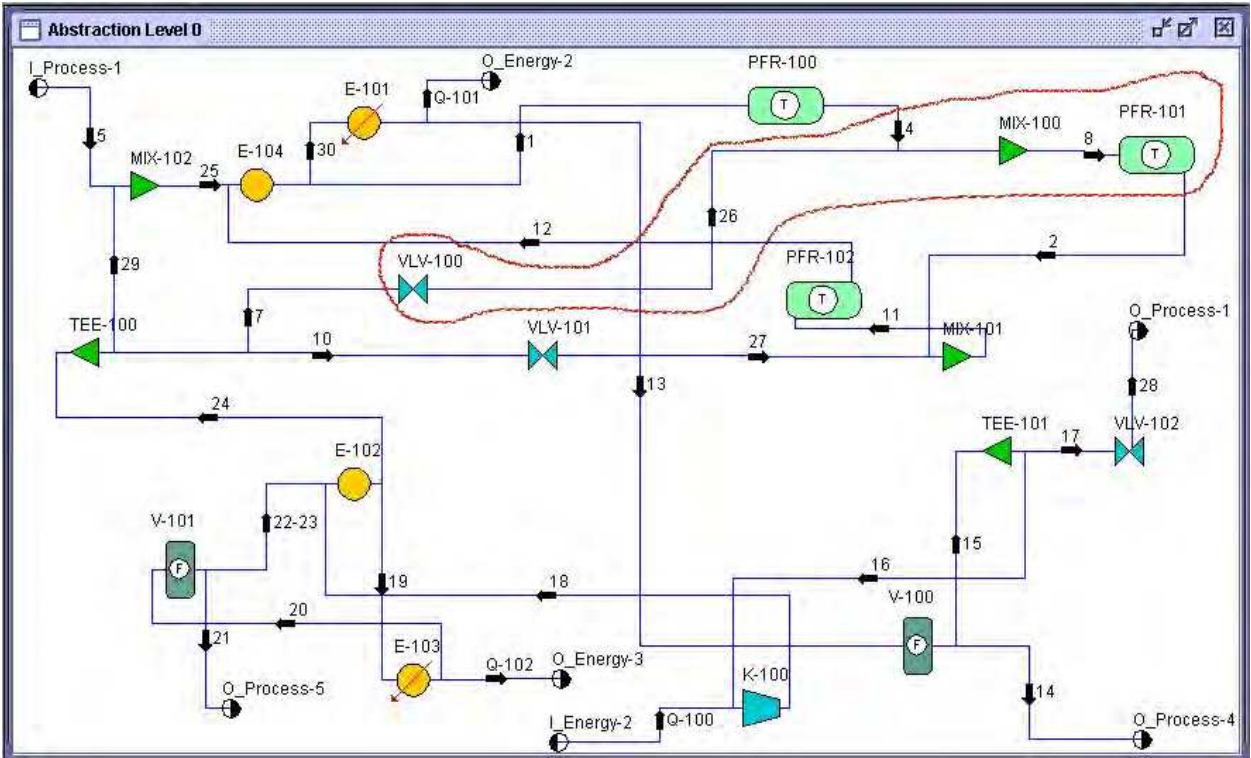


Fig. 13. Units composing the meta-reactor-3.

conditions are propagated to the units connected to the meta-reactor-3 in the flow path. The analysis in backward/forward stream directions finishes when a closer primary function is reached.

The state of the meta-reactor-3 is set to low capacity because the production of the main product is not enough. Since this meta-unit is not an initial unit in the path, its state may be originated by the effect of the performance of other units. Then back units are analysed. Considering the stream-4, the function to analyse is a source (PFR-100), which directly affects the concentration variable. Perhaps other back units in the same direction may affect the variable, but this is the closer primary function affecting the concentration variable. It may have low volume state, which originates the low capacity of meta-reactor-3. Therefore, the unit associated to this function is identified as cause unit and this branch of backward analysis in this direction finishes. Considering the stream-7, the function to analyse is a balance (TEE-100), which does not affect the variable. Then, the next function is analysed, which is a balance of temperature (E-102) that again does not affect directly the variable. The next function is storage (V-101), which affects the variable. This is another primary function affecting the concentration variable; it may have low volume state. Then its associated unit is a cause unit and this branch of backward analysis finishes.

Now, forward analysis is carried out following the output stream of the functional group. The low capacity state of meta-reactor-3 originates a low flow state and a low volume state, and consequently affects the following source functions producing a low capacity state in such function. Thus, considering the stream-2, the function to analyse is a balance (MIX-101), which does not affect the concentration variable. The next function is source (PFR-102), which affects the variable; it may have low capacity state originated for the low capacity state of the meta-reactor-3. Then, the unit associated with this function is a consequence unit. Since is the closer primary function affected in the forward stream direction, the forward analysis finishes.

Therefore, the identified cause and consequence units for meta-reactor-3 are shown in Table 3, which also represents the cause and consequence units for all the meta-units.

Candidate	Cause units	Consequence units
reactor-1 (PFR-100)	separator-2 (V-101)	reactor-2 (PFR-101)
reactor-2 (PFR-101)	reactor-1 (PFR-100)	reactor-3 (PFR-102)
reactor-3 (PFR-102)	reactor-2 (PFR-101)	separator-1 (V-101)
meta-reactor-1	reactor-1 (PFR-100) separator-2 (V-101)	reactor-3 (PFR-102)
meta-reactor-2	reactor-2 (PFR-101)	separator-1 (V-101)
meta-reactor-3	reactor-1 (PFR-100) separator-2 (V-101)	reactor-3 (PFR-102)
meta-reactor-4	reactor-2 (PFR-101)	separator-1 (V-101)
meta-reactor-5	separator-2 (V-101)	reactor-2 (PFR-101)
meta-reactor-6	separator-2 (V-101)	reactor-3 (PFR-102)
meta-reactor-7	separator-2 (V-101)	separator-1 (V-101)

Table 3. Cause and consequence units.

4.3 Generation of alternatives

A CBR module is used to obtain alternatives units/meta-units that may be adapted into the ammonia process. Again, the process representation showed in Figure 12 is used to denote the composition of meta-units; and the process representation showed in Figure 13 to denote the presence of units/meta-units in each abstraction level.

Assuming a threshold of 14 items, the most similar source cases from computing the global similarities are summarised in Table 4. It shows the percentage of similarity, the specific function, the inlet and outlet functions of the source case.

Rank	Similarity	Function	Inlet Function	Outlet Function
1	56%	meta-reactor	reaction	separation
2	43%	meta-reactor	reaction	flow change
3	37%	tubular reactor	tmp change	separation
4	31%	plug flow reactor	pres change	tmp change
5	30%	meta-reactor	separation	pres change
6	29%	tubular reactor	reaction	separation
7	29%	meta-reactor	tmp change	tmp change
8	27%	meta-reactor	pres change	separation
9	25%	tubular reactor	tmp change	tmp change
10	23%	tubular reactor	tmp change	separation
11	20%	plug flow reactor	flow change	pres change
12	20%	meta-reactor	pres change	tmp change
13	16%	meta-reactor	reaction	tmp change
14	15%	meta-reactor	flow change	tmp change

Table 4. Result of the global similarity computation for meta-reactor-3.

The modelling of the 50 processes generated 1590 cases in the case library. Therefore, the software prototypes were continually enhanced according to the needs of these processes.

The aspects considered in the evaluation of the framework were:

1. Modelling of the process
 - use of simplified models
 - suitable grouping of equipment/sections
 - intuitive goal-driven approach
 - comprehensive and clear representations of equipment/sections
 - easy and intuitive graphical interface

- transparent integration with the numerical simulator
- 2. Identification of candidates
 - clear and easy search over simple but consistent concepts
 - module easy to use
 - intuitive interpretation of results
- 3. Suggestion of equipment/sections
 - suggestions according to purpose-driven strategy
 - appropriate guidelines for modification/substitution
 - reuse of past design solutions
 - easy access to abstract and detailed data of proposed solutions
 - rapid response making agile the creation of alternative prototypes

5. Conclusions

This chapter describes a redesign support framework for technical processes based on hierarchical modelling. This hierarchical modelling is based on means-end and whole-parts aspects. The hierarchical representation enhances the reasoning mechanism to identify the elements to be modified and the possible alternatives. The framework focused on conceptual redesign issues where abstract models are employed. The processes are modelled hierarchically based on their functions and goals. The framework consists of four stages: design-description acquisition, identification of candidates to redesigned, generation of alternatives, and adaptation and evaluation of alternatives. The implementation of the framework was on the Chemical Engineering domain.

The redesign framework combines model-based reasoning and case-base reasoning techniques. This framework enables the designer to work directly with the conceptual design of an existing process (i.e. a process already in operation) to automatically generate abstract multiple-models, which can be modified to develop alternative process designs. The procedure can be seen as the reverse engineering approach of "replay and modify". This model-based approach provides an appropriate way of combining hierarchical and functional modelling to represent and reason about complex technical processes. The hierarchical case-based approach provides a systematic way of reusing the sections of previous processes.

The framework extends the use of Multimodelling and Multilevel Flow Modelling approaches to integrate mental abstract models about the behaviour of processes in the redesign activities. These models provide a more intuitive vision of reasoning on each task to be performed, and thus the redesign activities are enhanced.

The research has some limitations, some of the major ones are:

- The framework was implemented only in one domain. The ideas can be applied to another domain, but a new implementation will be necessary.
- The framework was tested with simulated plants. We did not have access to real plant information, but the results obtained were validated by a team of chemical engineers specialised in design of processes.
- The implementation of the framework is not manageable by novice users because important human designer decisions must be taken.

6. Acknowledgment

This research was partially funded by project number 175537 from “Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas”.

7. References

- (Aamodt 94) A. Aamodt & E. Plaza. Case-based reasoning: Foundational issues, methodological variations and system approaches. *AI Communications*, vol. 7, pages 39–59, 1994.
- (Arana 00) I. Arana H. Ahriz & P. Fothergill. Improving re-design support. In *Proceedings of the Fifth World Conference on Integrated Design and Process Technology (IDPT2000)*, 2000.
- (Arana 01) I. Arana H. Ahriz & P. Fothergill. *Redesign Knowledge Analysis, Representation and Reuse. Industrial Knowledge Management A Micro-level Approach*, 2001. Springer-Verlag.
- (Akin 82) O. Akin. Representation and architecture. Silver Spring, O. Akin and E.F. Weinelt eds., Maryland, Information Dynamics Inc., 1982.
- (Alberts 92) L.K. Alberts P.M. Wognum & N.J. Mars. Structuring design knowledge on the basis of generic components. *Artificial Intelligence in Design (AID'92)*, vol. In: Gero, J.S. (Ed.), pages 639–656, 1992. Kluwer, Dordrecht.
- (Alberts 93a) L.K. Alberts. YMIR: An ontology for engineering design. PhD thesis, University of Twente, 1993. Netherlands.
- (Alberts 93b) L.K. Alberts R.R. Bakker D. Beekman & P.M. Wognum. Model-based redesign of technical systems. *Proceedings of the 4th international workshop on principles of diagnosis*, 1993.
- (Bakker 94) R.R. Bakker S.J. van Eldonk P.M. Wognum & N.I. Mars. The use of model-based diagnosis in redesign. In *Proc. Reasoning about physical systems. 11th European Conference on Artificial Intelligence*, pages 647–651, A. Cohn Ed., 1994.
- (Ball 92) N.R. Ball & F. Bauert. The Integrated Design Framework: supporting the design process using a blackboard system. *Artificial Intelligence in Design (AID'92)*, vol. Gero, J.S. (Ed.), pages 327–348, 1992. Kluwer academic publishers.
- (Berker 96) I. Berker & D.C. Brown. Conflicts and Negotiation in Single Function Agent Based Design Systems. *Concurrent Engineering: Research and Applications, Journal, Special Issue: Multi Agent Systems in Concurrent Engineering*, vol. 4, no. 1, pages 17–33, 1996. Brown, D.C., Landes, S.E. and Petrie, C.J. (Eds.), Technomic Publishing Inc.
- (Bertalanffy 50) L.von Bertalanffy. An Outline of General Systems Theory. *British Journal for the Philosophy of Science*, vol. 1, no. 2, 1950.
- (Bhatta 94) S. Bhatta A. Goel & S. Prabhakar. Innovation in Analogical Design: A Model-Based Approach. *Proceedings of Artificial Intelligence in Design*, pages 57–74, 1994. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- (Bo 99) Y. Bo & F. Salustri. *Function Modeling Based on Interactions of Mass, Energy and Information*. Florida Artificial Intelligence Research Society, 1999.
- (Borner 96) K. Borner E. Pipping E. Tammer & C. Coulon. Structural Similarity and Adaptation. *Proceedings of the Third European Workshop on Case-Based Reasoning*, pages 58–75, 1996. Springer-Verlag.

- (Bras 92) B.A. Bras. Foundation for designing decision-based design processes. PhD thesis, University of Houston, 1992.
- (Brazier 94) F.M. Brazier P.H. Van Langen Zs. Ruttkay & J. Treur. On formal specification of design tasks. Artificial Intelligence in Design (AID'94), vol. Gero, J.S. and Sudweeks, F. (Eds.), pages 535–552, 1994. Kluwer Academic Publishers, Dordrecht.
- (Brazier 96) F.M. Brazier P.H. Van Langen J. Treur & N.J. Wijnngaards. Redesign and reuse in compositional knowledge-based systems. Knowledge Based Systems, Special Issue on Models and Techniques for Reuse of Designs, vol. 9, no. 2, pages 105–119, 1996.
- (Bridge 97) C. Bridge. From design to redesign: Revisiting design models to highlight similarities and differences. Seminar Presentation, 1997. Department of Architectural and Design Science, University of Sydney, Australia.
- (Brown 89) D.C. Brown & B. Chandrasekaran. Design problem solving: knowledge structures and control strategies. Morgan Kaufmann, 1989.
- (Brown 97) D.C. Brown & W.P. Birmingham. Understanding the Nature of Design. IEEE Expert, vol. 12, no. 2, pages 14–16, 1997.
- (Brown 98) D.C. Brown. Intelligent Computer-Aided Design. AI in Design Group, Computer Science Department, WPI, September 1998. Worcester, MA, USA.
- (Buckingham 91) S. Buckingham Shum. Cognitive Dimensions of Design Rationale. In People and Computers VI: Proceedings of HCI'91, pages 331–344, Cambridge University Press. Cambridge, 1991.
- (Chang 2003) Chang WC, Van YT. Researching design trends for the redesign of product form. Design Studies 24(2):173–180.
- (Chandrasekaran 90) B. Chandrasekaran. Design problem solving: a task analysis. AI Magazine, vol. 11, no. 4, pages 59–71, 1990.
- (Chandrasekaran 93) B. Chandrasekaran A. Goel & Y. Iwasaki. Functional Representation as Design Rationale. IEEE Computer, no. Special Issue on Concurrent Engineering, pages 48–56, 1993.
- (Chandrasekaran 00) B. Chandrasekaran & J.R. Josephson. Function in Device Representation. Engineering with Computers, Special Issue on Computer Aided Engineering, vol. 16, pages 162–177, 2000.
- (Checkland 81) P. Checkland. Systems thinking, systems practice. John Wiley and Sons, 1981.
- (Chen 2009) Li, S. and Chen, L., 2009, Pattern-based Reasoning for Rapid Redesign: A Proactive Approach. Research in Engineering Design, Vol. 21, pp. 25–42.
- (Chittaro 93) L. Chittaro G. Guida C. Tasso & E. Toppano. Functional and Teleological Knowledge in the Multimodeling Approach for Reasoning about Physical Systems: A case study in diagnosis. IEEE Transactions on Systems Man. and Cybernetics., vol. 23, no. 6, pages 1718–1751, 1993.
- (Chittaro 98) L. Chittaro & A.N. Kumar. Reasoning about function and its applications to engineering. Artificial Intelligence in Engineering, vol. 12, pages 331–336, 1998.
- (Culley 99) S.J. Culley. Final Report -Future Issues For Design Research Workshop. Technical Report, Faculty of Engineering and Design, University of Bath, 1999.
- (Das 94) D. Das S. Gupta & D. Nau. Reducing setup cost by automated generation of redesign suggestions. Proc. ASME Computers in Engineering Conference, pages 159–170, 1994.

- (Daube 89) F. Daube & B. Hayes-Roth. A Case-Based Mechanical Redesign System. In Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89) Vol. 2, pages 1402–1407, Morgan Kaufmann Publishers, 1989.
- (de Kleer 79) J. de Kleer. Causal and Teleological Reasoning in Circuit Recongnition. PhD thesis, Massachusetts Institute of Technology, 1979.
- (de Silva Garza 96) A. Gomez de Silva Garza & M.L. Maher. Design by interactive exploration using memory based techniques. Knowledge based systems, vol. 9, no. 1, pages 151–161, 1996.
- (DeLoach 04) S.A. DeLoach. The MaSE Methodology. Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook Series : Multiagent Systems, Artificial Societies and Simulated Organizations, vol. 11, 2004. Bergenti, F., Gleizes, M.P., Zambonelli, F. (Eds.) Kluwer Academic Publishing.
- (Dixon 89) J.R. Dixon M.J. Guenette R.K. Irani E.H. Nielsen M.F. Orelup & R.V. Welch. Computer-based models of design processes: the evaluation of design for redesign. NSF Engineering Design Research Conference, pages 491–506, 1989. University of Massachusetts, USA.
- (Douglas 88) J.M. Douglas. Conceptual design of chemical processes. Mc Graw Hill, New York, 1988.
- (Dove 2001) Dove R. Response ability: the language, structure, and culture of the agile enterprise. Wiley, New York.
- (Dunskus 95) B.V. Dunskus D.L. Grecu D.C. Brown & I. Berker. Using Single Function Agents to Investigate Con.ict. Artificial Intelligence in Engineering Design and Manufacturing (AIEDAM), Special Issue: Conflict Management in Design, vol. 9, no. 4, pages 299–312, 1995.
- (Eldonk 96) S.J. van Eldonk L.K. Alberts R.R. Bakker F. Diker & P.M. Wognum. Redesign of technical systems. Knowledge-based Systems, Special Issue on Models and Techniques for Reuse of Designs, vol. 9, pages 93–104, 1996.
- (Falkenhainer 91) B. Falkenhainer & K.D. Forbus. Compositional Modeling: Finding the Right Model for the Job. Artificial Intelligence, vol. 51, pages 95–143, 1991.
- (Falkenhainer 92) B. Falkenhainer & K.D. Forbus. Composing task-specific models. Automated Modelling, ASME, vol. 41, pages 1–9, 1992.
- (Fensel 01b) D. Fensel. Ontologies: A silver bullet for knowledge management and electronic commerce. Heidelberg, Germany, 2001.
- (Fischer 87) G. Fischer A.C. Lemke & C. Rathke. From design to redesign. In Proceedings of the 9th International Conference on Software Engineering, pages 369–376, IEEE Computer Society Press, 1987.
- (Fischo. 78) B. Fischo. P. Slovic & S. Lichtenstein. Fault trees: Sensitivity of estimated failure probabilities to problem representation. Journal of Experimental Psychology: Human Perception and Performance, vol. 4, 1978.
- (Forbus 84) K.D. Forbus. Qualitative Process Theory. Artificial Intelligence, vol. 24, pages 85–168, 1984.
- (Forster 96) J. Forster I. Arana & P. Fothergill. Redesign knowledge representation with DEKLARE. In Proceedings of KEML'96: 6th Workshop on Knowledge Engineering: Methods and Languages, Paris, France, 1996.

- (Forster 97b) J. Forster P. Fothergill & I. Arana. Enabling intelligent variant design using constraints. IEE Intelligent Design Systems Colloquium, vol. Digest No: 97/016, 1997. London.
- (Fothergill 95) P. Fothergill J. Forster J. A. Lacunza F. Plaza & I. Arana. DEKLARE: A methodological approach to re-design. Proceedings of Conference on Integration in Manufacturing, pages 109–122, 1995. K. R. von Barisani, P. A. MacConaill and K. Tierney (Eds.), Vienna. IOS Press.
- (Franke 92) D.W. Franke. A theory of teleology. PhD thesis, University of Texas at Austin, 1992.
- (French 85) R. French & J. Mostow. Toward Better Models of the Design Process. AI Magazine, vol. 6, no. 1, pages 44–57, 1985.
- (French 93) M.J. French R.V. Chaplin & P.M. Langdon. A creativity aid for designers. International Conference on Engineering Design, ICED'93, pages 53–59, 1993. The Hague, Netherlands.
- (Fricke 2005) Fricke E, Schulz A P. Design for changeability (DfC): principles to enable changes in systems throughout their entire lifecycle. Systems Engineering 8(4):342–359. Wiley Periodicals.
- (Gero 90a) J.S. Gero. Design prototypes: a knowledge representation schema for design. AI Magazine, vol. 11, no. 4, pages 26–36, 1990.
- (Gero 90b) J.S. Gero & A. Rosenman. A conceptual framework for knowledge-based de-sign research at Sydney University's design computing unit. Artificial Intelligence in Engineering, vol. 5, no. 2, pages 65–77, 1990.
- (Gero 04) J.S. Gero & U. Kannengiesser. The situated Function-Behaviour-Structure framework. Design Studies, vol. 25, no. 4, pages 373–391, 2004.
- (GIA 04) GIA. Greener Industry Association: Ammonia Process.
http://www.uyseg.org/greener_industry/index.htm, 2004.
- (Goel 89) A. Goel & B. Chandrasekaran. Functional representation of designs and re-design problem solving. Proc. Eleventh International Joint Conference on Artificial Intelligence, pages 1388–1394, 1989. Los Altos, California: Morgan Kaufmann Publishers.
- (Goel 91) A. Goel. A model-based approach to case adaptation. In In Proceedings of the 13th Annual Conference of the Cognitive Science Society (CogSci'91), pages 143–148, Chicago, Illinois, 1991.
- (Goel 92) A. Goel & B. Chandrasekaran. Case-Based Design: A Task Analysis. Artificial Intelligence Approaches to Engineering Design, Innovative Design, vol. II, no. 6, pages 165–184, 1992. Academic Press, Tong y D. Sriram (editors).
- (Goel 94b) A. Goel & S. Prabhakar. A control architecture for redesign and design veri-fication. Proc. Second Australian and New Zealand Conference on Intelligent Information Systems Conference, pages 377–381, 1994. Brisbane, Qld., Australia.
- (Goel 97a) A. Goel A. Gomez de Silva Garza N. Grue J.W. Murdock & M.M. Recker. Functional Explanations in Design. IJCAI-97 Workshop on Modeling and Reasoning, 1997.
- (Gomez-Perez 04) A. Gomez-Perez R. Gonzalez-Cabero & M. Lama. Development of Semantic Web Services at the Knowledge Level. In The European Conference on Web Service (ECOWS'04), pages 72–86, 2004.

- (Grossmann 00) I.E. Grossmann & A.W. Westerberg. Research Challenges in Process Systems Engineering. *AIChE Journal*, vol. 46, no. 9, pages 1700–1703, 2000.
- (Gruber 93) T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, vol. 5, no. 2, pages 199–220, 1993.
- (Gundersen 90) T. Gundersen. Retrofit Process Design -Research and Applications of Systematic Methods. *Foundations of Computer-Aided Process Design*, pages 213–240, 1990. J.J. Sirola, I. E. Grossmann and G. Stephanopoulos (Eds.).
- (Han 95) Ch. Han J.M. Douglas G. Stephanopoulos. Agent-Based Approach to a Design Support System For the Synthesis of Continuous Chemical Processes. *Computers in Chemical Engineering*, vol. 19 (Supplement), pages S63–S69, 1995.
- (Heo 98) D.H. Heo A.C. Parker & C.P. Ravikumar. An Evolutionary Approach to System Redesign. *Proc. Eleventh International Conference on VLSI Design*, pages 359–362, 1998. Chennai, India.
- (Hertwig 01) T.A. Hertwig A. Xu A.B. Nagy R.W. Pike J.R. Hopper & C.L. Yaws. A prototype system for economic, environmental and sustainable optimization of a chemical complex. *Proc. 11th European Symposium on Computer Aided Process Engineering*, pages 1017–1022, 2001. R. Gani and S.B. Jorgensen eds, Elsevier.
- (Hoover 91) S.P. Hoover J.R. Rinderle S. Finger. Models and abstractions in design. In *Proceedings of the International Conf. on Engineering Design, ICED'91*, pages 46–57, Zurich, 1991.
- (Howe 86) A. Howe P. Cohen J. Dixon & M.D. Simmons. A domain-independent program for mechanical engineering design. In *Proceedings of the 1st International Conference on Applications of AI in Engineering Problems*, pages 289–299, Southampton University, U.K., 1986.
- (Hysys 04) Hysys. Introduction to hysys.plant ver. 3.1. AEA Technology Engineering Software, Hyprotech, Calgary, Canada, 2004.
- (JESS 08) JESS. Jess 7.1 manual. <http://herzberg.ca.sandia.gov/jess/docs/index.shtml>, 2008.
- (Keuneke 91) A.M. Keuneke. Device Representation: The Significance of Functional Knowledge. *IEEE Expert*, vol. 6, no. 2, pages 22–25, 1991.
- (Kim 93) G.Y. Kim & G. Bekey. Construting Design Plans for DFA Redesign. *Proc. IEEE International Conference on Robotics and Automation*, pages 312–318, 1993.
- (Kirkwood 88) R.L. Kirkwood M.H. Locke & J.M. Douglas. A prototype expert system for synthesizing chemical process flowsheets. *Comp. Chem. Eng.*, vol. 12, no. 4, pages 329–343, 1988.
- (Kitamura 98) Y. Kitamura & R. Mizoguchi. Functional Ontology for Functional Understanding. *Proceedings of The Twelfth International Workshop on Qualitative Reasoning*, AAAI Press, pages 77–87, 1998. Cape Cod, USA.
- (Kitamura 99) Y. Kitamura & R. Mizoguchi. Towards Redesign based on Ontologies of Functional Concepts and Redesign Strategies. In *Second International Workshop On Strategic Knowledge And Concept Formation*, page JSPS8, Iwate, Japan, 1999.
- (Kolodner 93) J. Kolodner. Case-based reasoning. Morgan Kaufman Publishers, Inc., 1993.
- (Kraslawski 00) R. Ben-Guang H. Fan-Yu A. Kraslawski & L. Nystrom. Review: study on the methodology for retrofitting chemical processes. *Chemical Engineering Technology*, vol. 23, no. 6, pages 479–484, 2000.

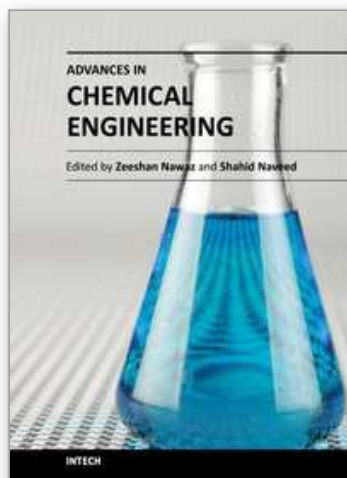
- (Kuraoka 03) K. Kuraoka & R. Batres. An Ontological Approach to Represent HAZOP Information. Technical Report TR-2003-01, Process Systems Engineering Laboratory, Tokyo Institute of Technology, 2003.
- (Lander 97) S.E. Lander. Issues in Multi-agent Design Systems. *IEEE Expert*, vol. 12, no. 2, pages 18–26, 1997.
- (Leveson 00) N.G. Leveson. Intent specifications: an approach to building human-centered specifications. *IEEE Transactions on Software Engineering*, vol. 26, no. 1, pages 15–35, 2000.
- (Lind 90) M. Lind. Representing Goals and Functions of Complex Systems. Technical Report 90-D-38, Institute of Automatic Control Systems, Technical University of Denmark, 1990.
- (Lind 94) M. Lind. Modelling goals and functions of complex industrial plants. *Applied Artificial Intelligence*, vol. 8, pages 259–284, 1994.
- (Lind 96) M. Lind. Status and challenges of intelligent plant control. *Annual Reviews in Control*, vol. 20, pages 23–41, 1996.
- (Lind 99) M. Lind. Plant Modeling for Human Supervisory Control. *Transactions of the Institute of Measurement and Control*, vol. 21, no. 4/5, pages 171–180, 1999.
- (Linnho. 88) B. Linnho. G.T. Polley & V. Sahdev. General Process Improvements Through Pinch Technology. *Chem. Eng. Progr.*, vol. 84, pages 51–58, 1988.
- (Maher 95) M.L. Maher B. Balachandran & D.M. Zhang. Case-based reasoning in design. Lawrence Erlbaum Associates, Sydney, 1995.
- (Maher 97a) M.L. Maher & A. Gomez de Silva Garza. Case-Based Reasoning in Design. *IEEE Expert*, vol. 12, no. 2, pages 34–41, 1997.
- (Maher 97b) M.L. Maher S. Simno. & J. Mitchell. Formalising building requirements using an activity/space model. *Automation in construction*, vol. 6, pages 77–95, 1997.
- (Maher 01) M.L. Maher & A. Gomez de Silva Garza. GENCAD: A Hybrid Analogical/Evolutionary Model of Creative Design. *Proc. of the 4th International Conference on Computational Models of Creative Design*, 2001. J.S. Gero and M.L. Maher (eds).
- (Maulik 92) P.C. Maulik M.J. Flynn D.J. Allstot & L.R. Carley. Rapid Redesign of Analog Standard Cells Using Constrained Optimization Techniques. *Proc. IEEE Custom Integrated Circuits Conference*, pages 8.1.1–81.3, 1992.
- (Mitchell 83) T.M. Mitchell L.I. Steinberg S. Kedar-Cabelli V.E. Kelly J. Shul-Man & T. Weinrich. An intelligent aid for circuit redesign. In *Proceedings of the 3th National Conference on Artificial Intelligence (AAAI-83)*, Washington, USA, 1983.
- (Mostow 89) J. Mostow. Design by derivational analogy: Issues in the automated replay of design plans. *Artificial Intelligence*, vol. 40, no. 1-3, pages 119–184, 1989.
- (Nelson 90) D.A. Nelson & J.M. Douglas. A systematic procedure for retrofitting chemical plants to operate utilising different reaction paths. *Ind. Eng. Chem. Res.*, vol. 29, pages 819–829, 1990.
- (Niles 01) I. Niles & A. Pease. Towards a Standard Upper Ontology. In *2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 2–9, 2001.
- (Ohsuga 97) S. Ohsuga. Strategic Knowledge Makes Knowledge Based Systems Truly Intelligent. In *Proceedings of the First International Workshop on Strategic Knowledge and Concept Formation*, pages 1–24, Lutchi Research Centre, 1997.

- (Ozone 08) Ozone. Ozone Developers' Guide.
<http://www.ozone-db.org/frames/documentation/overview.html>, 2008.
- (Papoulias 83) S.A. Papoulias & I.E. Grossman. A Structural Optimization Approach in Process Synthesis II: Heat Recovery Networks. *Computers and Chemical Engineering*, vol. 7, pages 707–721, 1983.
- (Pasanen 01) A. Pasanen. Phenomenon-Driven Process Design Methodology. PhD thesis, Department of Process and Environmental Engineering of the University of Oulu, Finland, 2001.
- (Pistikopoulos 87) E.N. Pistikopoulos & I.E. Grossman. Optimal Retrofit Design for Improving Process Flexibility in Linear Systems. Technical Report EDRC-06-24-87, Engineering Design Research Center. Carnegie Mellon University, 1987.
- (Pos 97) A. Pos. Problem Solving for Redesign. 10th European Knowledge Workshop on Knowledge Acquisition. Modeling and Management, pages 205–220, 1997. Sant Feliu de Guíxols, Catalonia.
- (Price 97) C.J. Price I.S. Pegler M.B. Ratcliffe & A. McManus. From troubleshooting to process design: closing the manufacturing loop. In 2nd International Conference on Case-based Reasoning Research and Development, pages 114–121, London, UK, 1997. LNCS Vol. 1266, Springer-Verlag.
- (Price 98) C.J. Price. Function-directed electrical design analysis. *Artificial Intelligence in Engineering*, vol. 12, pages 445–456, 1998. Elsevier.
- (Price 03) C.J. Price N.A. Snooke & S.D. Lewis. Adaptable Modeling of Electrical Systems. In 17th International Workshop on Qualitative Reasoning, pages 147–153, 2003. Brasilia, Brazil.
- (Qian 92) L. Qian & J.S. Gero. A design support system using analogy. *Proceedings of the Second International Conference on AI in Design*, pages 795–813, 1992. Kluwer Academic Publishers.
- (Rapoport 94) H. Rapoport R. Lavie & E. Kehat. Retrofit design of new units into an existing plant: Case study: Adding new units to an aromatics plant. *Computers and Chemical Engineering*, vol. 18, no. 8, pages 743–753, 1994.
- (Rasmussen 85) J. Rasmussen. The Role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 2, 1985.
- (Rasmussen 86) J. Rasmussen. Information processing and human-machine interaction: An approach to cognitive engineering. North Holland, 1986.
- (Salomons 95) O.W. Salomons. Computer Support in the Design of Mechanical Products. PhD thesis, University of Twente, 1995.
- (Sasajima 95) M. Sasajima Y. Kitamura M. Ikeda & R. Mizoguchi. FBRL: A Function and Behavior Representation Language. *Proceedings of IJCAI*, pages 1830–1836, 1995.
- (Schoen 91) E. Schoen. Intelligent Assistance for the Design of Knowledge based Systems. PhD thesis, Stanford University, 1991.
- (Sembugamoorthy 86) V. Sembugamoorthy & B. Chandrasekaran. Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems. *Experience, Memory and Reasoning*, J.L. Kolodner and C.K. Riesbeck, eds, Lawrence Erlbaum, Hillsdale, N.J., pages 47–73, 1986.
- (Sylvester 00) R.W. Sylvester W.D. Smith & J. Carberry. Information and modelling for greener process design. *AIChE Symp. Series*, vol. 96, no. 323, pages 26–30, 2000.

- (Simon 96) H.A. Simon. The sciences of the artificial. MIT Press, Cambridge, Massachusetts, 1996.
- (Skuce 93) D. Skuce. A multi functional knowledge management system. Knowledge Acquisition, vol. 5, pages 305–346, 1993.
- (Smith 87) R. Smith & B. Linnho. Process integration using pinch technology. Proceedings of ATEE Symp. Energy Management in Industry, 1987. Paris.
- (Smyth 96) B. Smyth & M.T. Keane. Using adaptation knowledge to retrieve and adapt design cases. Knowledge-based Systems, Special Issue on Models and Techniques for Reuse of Designs, vol. 9, no. 2, pages 127–136, 1996.
- (Smyth 01) B. Smyth M.T. Keane & P. Cunningham. Hierarchical Case-Based Reasoning Integrating Case-Based and Decompositional Problem-Solving Techniques for Plant-Control Software Design. IEEE Trans. on Knowledge and data engineering, vol. 13, no. 5, pages 793–812, 2001.
- (Sowa 95) J. Sowa. Distinctions, Combinations and Constraints. In Proceedings of IJCAI 95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995.
- (Steier 91) D. Steier. Automating Algorithm Design within a General Architecture for Intelligence. Automating Software Design, pages 577–602, 1991. Lowry, M.R. and McCartney, R.D. (Eds.).
- (Steinberg 85) L. Steinberg & T. Mitchell. The Redesign System: A Knowledge-base approach to VLSI CAD. IEEE Design and Test of Computers, vol. 2, no. 45-54, 1985.
- (Stephanopoulos 90a) G. Henning G. Stephanopoulos & H. Leone. MODEL.LA A modelling framework language for process engineering II. Multifaced modelling of processing systems. Computers and Chemical Engineering, vol. 14, no. 8, pages 847–869, 1990.
- (Stephanopoulos 90b) G. Stephanopoulos G. Henning & H. Leone. MODEL.LA A mod-elling framework language for process engineering I. The formal framework. Computers and Chemical Engineering, vol. 14, no. 8, pages 813–846, 1990.
- (Stroulia 92a) E. Stroulia M. Shankar A. Goel & L. Penberthy. A Model-Based Approach to Blame Assignment in Design. Proceedings of the Second International Conference on AI in Design, pages 519–537, 1992. J.S. Gero (editor).
- (Stroulia 92b) E. Stroulia & A. Goel. Generic Teleological Mechanisms and their use in Case Adaptation. Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society, pages 319–324, 1992. Indiana, USA.
- (Struss 91) P. Struss. A theory of model simplification and abstraction for diagnosis. In Proceedings of the Fifth International Workshop on Qualitative Reasoning, pages 25– 57, 1991.
- (Subba-Rao 99) S. Subba-Rao A. Nahm Z. Shi X. Deng & A. Syamil. Artificial intelligence and expert systems applications in new product development-a survey. Journal of Intelligent Manufacturing, vol. 10, no. 3-4, pages 231–244, 1999. Kluwer Academic Publishers.
- (Sumi 97) Y. Sumi. Supporting the Acquisition and Modelling of Requirements in Software Design. In Proceedings of the First International Workshop on Strategic Knowledge and Concept Formation, pages 205–216, Lutchi Research Centre, 1997.
- (Sun 08) Sun Microsystems. Java Language Specification. <http://java.sun.com/docs/index.html>, 2008.

- (Sycara 92) K. Sycara. CADET: A case-based synthesis tool for engineering design. *International Journal for Expert Systems*, no. 2, pages 157–188, 1992.
- (Takeda 90a) H. Takeda P. Verkaamp T. Tomiyama & H. Yoshikawa. Modelling Design Processes. *AI Magazine*, vol. 11, no. 4, pages 37–48, 1990.
- (Takeda 94a) H. Takeda. Abduction for design. In *Proceedings of Formal Design Methods for CAD*, pages 221–243, Tallinn, Estonia, 1994.
- (Tay 2003) Tay FEH, Gu J. A methodology for evolutionary product design. *Engineering with Computers* 19:160-173.
- (Thornton 93) A.C. Thornton & A. Johnson. Constraint specification and satisfaction in embodiment design. In *Proceedings of the International Conference on Engineering Design, ICED 93*, pages 1319–1326, The Hague, 1993.
- (Tjoe 86) T.N. Tjoe & B. Linnho. Using pinch technology for processes retrofit. *Chem. Eng.*, vol. 93, pages 47–60, 1986.
- (Tomiyama 87) T. Tomiyama & H. Yoshikawa. Extended General Design Theory. In *Proceedings of the IFIP WG 5.2 Working Conference on Design Theory for CAD*, pages 95–125, 1987.
- (Treur 89) J. Treur. A logical analysis of design tasks for expert systems. *International Journal of Expert Systems*, vol. 2, pages 233–253, 1989.
- (Turton 98) R. Turton R.C. Bailie W.B. Whiting & J.A. Shaeiwitz. Analysis, synthesis and design of chemical processes. Prentice-Hall, New Jersey, 1998.
- (Uerdingen 01) E. Uerdingen U. Fischer & K. Hunderbuhler. A screening method for identifying economic improvement potentials in retrofit design. In *ESCAPE-11 (European Symposium on Computer Aided Process Engineering)*, pages 573–578, 2001.
- (Ullman 91) D.G. Ullman. Design histories: archiving the evolution of products. In *Proceedings of the DARPA Workshop on Manufacturing*, Salt Lake City, USA, 1991.
- (Umeda 90) Y. Umeda T. Tomiyama & H. Yoshikawa. Function, Behaviour and Structure. *Applications of Artificial Intelligence in Engineering V*, pages 177–193, 1990. J.S. Gero (Ed).
- (Umeda 92) Y. Umeda T. Tomiyama & H. Yoshikawa. A design methodology for a self-maintenance machine based on functional redundancy. *Design Theory and Methodology -DTM'92*, pages 317–324, 1992.
- (Umeda 94) Y. Umeda T. Tomiyama H. Yoshikawa & Y. Shomimura. Using Functional maintenance to improve fault tolerance. *IEEE Expert*, vol. 9, no. 3, pages 25–31, 1994.
- (Umeda 97) Y. Umeda & T. Tomiyama. Functional Reasoning in Design. *IEEE Expert*, vol. 12, no. 2, pages 42–48, 1997.
- (Vaselenak 87) J.A. Vaselenak I.E. Grossman & A.W. Westerberg. Optimal Retrofit Design of Multiproduct Batch Plant. *Industrial and Engineering Chemistry Research*, vol. 26, no. 4, pages 718–726, 1987.
- (Vescovi 93) M. Vescovi & Y. Iwasaki. Device design as functional and structural re-ment. In *Working Notes of the IJCAI'93 Workshop on AI in Design*, pages 55–60, 1993.
- (Vicente 92) K.J. Vicente & J. Rasmussen. Ecological interface design: Theoretical foundations. *IEEE Trans. on Systems, Man and Cybernetics*, vol. 22, no. 4, 1992.
- (Wetzel 00) B. Wetzel. Selection Engine. <http://selectionengine.sourceforge.net/>, 2000.

- (Wielinga 97) B.J. Wielinga & A.Th. Schreiber. Configuration-design problem solving. *IEEE Expert*, vol. 12, no. 2, pages 49-56, 1997.
- (Wood 01) M. Wood & S.A. DeLoach. An Overview of the Multiagent Systems Engineering Methodology. *Agent-Oriented Software Engineering, LNAI*, vol. 1957, 2001. P. Ciancarini, M. Wooldridge, (Eds.).
- (Xin 01) W. Xin & X. Guangleng. Supporting design reuse based on integrated design rationale. *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pages 1909-1912, 2001.
- (Yoshikawa 91) H. Yoshikawa F. Arbab & T. Tomiyama. *Intelligent CAD III: Selected and Reviewed Papers and Reports*. In *Third International Workshop on Computer Aided Design*, 1991.



Advances in Chemical Engineering

Edited by Dr Zeeshan Nawaz

ISBN 978-953-51-0392-9

Hard cover, 584 pages

Publisher InTech

Published online 23, March, 2012

Published in print edition March, 2012

Chemical engineering applications have been a source of challenging optimization problems in terms of economics and technology. The goal of this book is to enable the reader to get instant information on fundamentals and advancements in chemical engineering. This book addresses ongoing evolutions of chemical engineering and provides overview to the state of the art advancements. Molecular perspective is increasingly important in the refinement of kinetic and thermodynamic modeling. As a result, much of the material was revised on industrial problems and their sophisticated solutions from known scientists around the world. These issues were divided into two sections, fundamental advances and catalysis and reaction engineering. A distinct feature of this text continues to be the emphasis on molecular chemistry, reaction engineering and modeling to achieve rational and robust industrial design. Our perspective is that this background must be made available to undergraduate, graduate and professionals in an integrated manner.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ivan Lopez-Arevalo, Victor Sosa-Sosa and Saul Lopez-Arevalo (2012). Modelling Approach for Redesign of Technical Processes, *Advances in Chemical Engineering*, Dr Zeeshan Nawaz (Ed.), ISBN: 978-953-51-0392-9, InTech, Available from: <http://www.intechopen.com/books/advances-in-chemical-engineering/modelling-approach-for-redesign-of-technical-processes>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen