DESIGN METHODOLOGY FOR MECHATRONIC SYSTEMS

AN APPROACH USING FUNCTION/MEANS TREE AND CHANNEL/AGENCY NET

VICTOR JULIANO DE NEGRI
KAROL MUÑOZ SALAS
VINÍCIUS VÍGOLO

Victor Juliano De Negri
Karol Muñoz Salas
Vinícius Vígolo

Design Methodology for Mechatronic Systems:

An Approach Using Function/Means Tree and Channel/Agency Net

Florianópolis
2025

Catalogação na fonte pela Biblioteca Universitária
da Universidade Federal de Santa Catarina

D392d De Negri, Victor Juliano
Design methodology for mechatronic systems [recurso eletrônico] : an approach using function, means tree and channel, agency net / Victor Juliano De Negri, Karol Muñoz Salas, Vinícius Vígolo. – Florianópolis : UFSC, 2025.
111 p : il., fots.

E-book (PDF)

ISBN 978-85-8328-351-5

1. Mecatrônica. 2. Mecatrônica – Projetos – Metodologia. I. Muñoz Salas, Karol. II. Vígolo, Vinícius. III. Título.

CDU: 621

Elaborada pela bibliotecária Dênira Remedi – CRB-14/1396

UNIVERSIDADE FEDERAL DE SANTA CATARINA

EMC-UFSC LASHIP HYDRAULICS & PNEUMATICS

PUCP

The Current Maintainer of this work is Victor Juliano De Negri
Digital version, May 2025

AUTHORS

VICTOR JULIANO DE NEGRI

He received his Mechanical Engineering degree in 1983, from UNISINOS, Brazil, M. Eng. degree in 1987 and D. Eng. degree in 1996, both from UFSC, Brazil. In 2010 he took a sabbatical at the PTMC, University of Bath, UK. He is professor at the Mechanical Engineering Department at the Federal University of Santa Catarina (UFSC) and the Head of the Laboratory of Hydraulic and Pneumatic Systems (LASHIP). From 2017 to 2024 he was the Head of the SC2C.Aero Santa Catarina’s Center of Convergence for Aerospace Technologies. He is member of ABCM - Brazilian Society of Mechanical Sciences and Engineering and ASME - American Society of Mechanical Engineers. Member of the Board of Directors of GFPS - Global Fluid Power Society and associate editor of the International Journal of Fluid power. He has coordinated several research projects with industry and governmental agencies in the areas of hydraulic components, power generating plants, mobile hydraulics, pneumatic systems and positioning systems.

KAROL MUÑOZ SALAS

She received her Electronic Engineering degree in 2006 from the Pontificia Universidad Católica del Perú and a Master's degree in Automation and Systems Engineering from the Federal University of Santa Catarina in 2014. Since 2015, she has been working as a professor at the Pontificia Universidad Católica del Perú, teaching courses in automation and Industry 4.0, Computer-Integrated Manufacturing, and Design of Mechatronic System.

VINÍCIUS VIGOLO

He received his degree in Mechanical Production Engineering in 2016 from UNOESC, Brazil, and a Master's degree in Mechanical Engineering from the Federal University of Santa Catarina (UFSC) in 2018. From 2018 to 2021, he worked as a research engineer at UFSC on projects focused on developing a digital hydraulic actuator for aircraft applications and applying pneumatic technology to control hydraulic turbines. From 2020 to 2022, he served as an assistant professor at the Pontificia Universidad Católica del Perú, teaching the course Mechatronic Design Methodology. In 2023, he participated in a PhD exchange program at Technische Universität Dresden (Germany). Since 2016, he has been part of the LASHIP research group (Laboratory of Hydraulic and Pneumatic Systems), where he currently works as a researcher and is set to complete his Doctorate in Mechanical Engineering at UFSC in 2025.

SUMMARY

PREFACE

Most devices and machines supporting industry, agriculture and earthmoving, energy generation, urban services, and many other production and everyday applications are built using multiple technologies. Combination of hydraulic and pneumatic, mechanical and electrical principles with electronic hardware and software is very common, forming the so-called mechatronic systems.

Each technical domain has its own design practices and tools. However, the system parts will operate in an integrated manner and, consequently, a global view and a systematic integration of the subsystems is required throughout the design process.

A very well-structured design methodology is the one developed by Pahl & Beitz and also published as the VDI 2221 standard. It defines four main design phases, subdivided into steps and tasks that must be completed to achieve the final product design. This methodology is primarily intended for mechanical systems such that electronic and software development, as well as integration of technical domains, are not covered in depth.

A design methodology for mechatronic systems is outlined in VDI 2206 standard, where a procedural model is presented, establishing very general guidelines for the system integration and design in specific domains.

Another important approach that can support the structuring of mechatronic system design is software development, mainly derived from object-oriented software area. The UML – Unified Modeling Language and its extension, the SysML – Systems Modeling Language, have several concepts, definitions, and graphical models suitable for representing characteristics of mechatronic systems. However, the individual design of physical subsystems, such as mechanical, electrical, and hydraulic and pneumatics parts, is not addressed.

An important support for mechatronic systems designer is to have a clear model of the entire design process, starting from the main phases and breaking down into its stages, tasks, and activities. Therefore, by combining the principles, models, and tools from the three methodologies mentioned above with the fundamentals of design and implementation of control and automation system, this book presents a comprehensive model of the design process for mechatronic systems.

The content presented in this book comes from the authors' experience in systems modeling, mechatronic systems design, fluid power systems, and machine control and automation. Studies on these topics have been carried out since the 1990s at LASHIP – Hydraulic and Pneumatic

Systems Laboratory of the Federal University of Santa Catarina and applied in different areas, such as industrial equipment, hydroelectric plants, mobile machines, and aeronautics. The impetus for organizing this book came in 2019, from the invitation of the Coordinator of Mechatronic Engineering at PUCP – Pontifical Catholic University of Peru, Prof. Ericka Madrid Ruiz, with the support of the "International Professor - PUCP Academic Fund" program, to format a project methodology for structuring Mechatronic Engineering teaching at PUCP.

Victor J. De Negri
Karol Muñoz Salas
Vinícius Vigolo

1. INTRODUCTION

The growing complexity of mechatronic systems is evident, whether for market, security, or even comfort reasons. This scenario requires complete management by designer, not only in the phases that make up the design, but also in the technological fields involved. Therefore, there is a constant search for the establishment of systematic criteria and methods that aim to make the design process more manageable, making it possible to foresee consequences and implications arising from decisions taken throughout the project, reducing the possibility of unpleasant surprises at its end (Paes & De Negri, 2003).

Due to the expansion of man's knowledge about physical phenomena and the consequent diversity of technologies, grows the importance of a multidisciplinary approach for the analysis and, mainly, for the design of devices that meet the current needs. In this context, mechatronics – the integration of mechanical, electrical, and software engineering – has become an essential framework for developing highly efficient and intelligent systems capable of performing complex tasks with precision and reliability.

This book aims to provide a systematic approach to the design process of mechatronic systems. The methodology presented in the following chapters results from a thoughtful integration of two well-established design frameworks: the mechanical design process formulated by Pahl and Beitz and the mechatronic design methodology presented in VDI 2206. In addition, concepts and definitions from object-oriented software development are applied for the mechatronic area.

The progressive concatenation of functional decomposition and system materialization is accomplished using Function/Means tree diagrams, assisting in the development of creative and effective design solutions. To improve clarity and usability, this book uses Channel/Agency net notation to visually represent the key phases, steps, and tasks of the design process, along with Place/Transition Petri-Nets to model activity sequences. By combining classical design methods with modern tools, the resulting framework offers a clear and practical methodology for designing mechatronic systems. This integrated approach connects traditional and modern practices, providing a structured path to manage the complexity of developing advanced, integrated systems.

The book's content begins with fundamental concepts about systems and modeling, which are essential for designers to choose the appropriate representations at each design stage and for the technical domains, as well as for understanding the design process model. The following chapter presents the general structure of a mechatronic system, a central model used in all phases of the design. Next, a diagrammatic model for the design of mechatronic systems is presented, including a description of the main subtasks required in each technical area. At the end, an example of a design process is detailed.

2. FUNDAMENTALS OF SYSTEMS AND MODELS

2.1. DEFINITION OF TECHNICAL SYSTEMS

2.1.1. Introduction

Technology is used in the production of goods or services to meet the need of human beings. It includes machines, vehicles, computers, robots, printers, pieces of equipment, etc. In order to make reference to these devices in general, the term **technical system** can be used. As stated by Frederick and Carlson (1971), a technical system is "an organized collection of interacting units - possibly including men and machines - designed to achieve some objective or set of objectives through the manipulation and control of materials, energy, and information". The units that comprise a system can be called components or elements and they can be refined in other components or elements, and so on. Therefore, each part can be characterized as a system (Hubka & Eder, 1988).

Complementary, the definition of system given by Nygaard (1986) states that: "A system is a part of the world that a person (or group of persons) chooses to regard as a whole consisting of components, each component characterized by properties that are selected as being relevant and by actions related to these properties and those of other components. According to this definition, no part of the world "is a system" as an inherent property. It is a system if one chooses a system perspective."

Therefore, based on the statements above and other authors, a technical system can be denoted by the following aspects:

- It can be recognized through the identification of its units and the means of interaction;
- It exists to fulfill a certain function in the macrosystem of which it is a part;
- It acts on energy, matter, and information. Through these resources the interaction with the system user and other systems occurs;
- It must be observed through a perspective. A perspective, or a point-of-view, is based on a particular theory or fundamentals.

A technical system can be diagrammatically represented as shown in Figure 1. Energy, matter, and information characterize the general resources that will be consumed and produced as a result of the system operation. This operation results from the application of processes of change or transformation of resources such as transportation, expansion, conversion, transformation, decomposition, calculation, control, processing, etc.

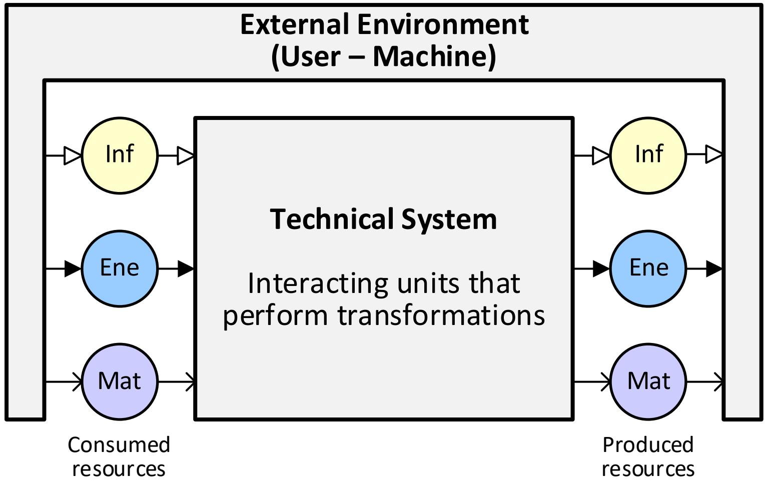


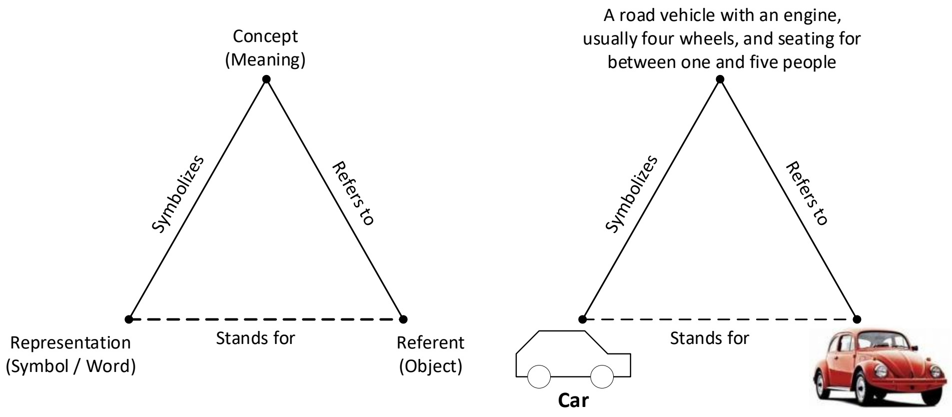
Figure 1 – General representation of a technical system

This representation uses the Channel/Agency Net notation (C/A Net), a generic diagrammatic model using a simple structure and easy to be assimilated by people from different backgrounds. This model will be used in this work to be describing the design process as well as to be representing the system under design. The C/A Net will be presented in Section 2.3.

2.1.2. Characterization of Information

As presented above, one of the general resources flowing throughout a system is **information**. A precise and objective definition of the term information comes from the triangle of meaning (or triangle of reference) shown in Figure 2. The left vertex is the symbol or word, the upper vertex is the concept, intention, thought, idea, or sense and the right vertex is the referent, object, or extension. Perception correlates objects to concepts and language correlates concepts to words (symbols), but the relationship between symbol and object is an indirect correlation, resulting from the direct correlations of perception and language.

Information consists of a symbol and the relation that it has to the object, that means, the indirect correlation in the dashed line shown in Figure 2. In other words, to have information about an object represented by a symbol, a meaning (concept) must be known. For example, considers the voltage in an electrical circuit as an object, but it is only recognized such as because the concept of voltage is known, that means, it corresponds to "the difference in electric potential between two points, which is defined as the work needed per unit of charge to move a test charge between the two points". A symbol "V" can be used to represent the actual voltage and carries information because it is understood the meaning of voltage. Another example can be a car (object), as shown in Figure 2b, symbolized by either a drawing or word "Car". Someone understands these symbols because he/she learned the definition presented in the upper vertex of the triangle and, consequently, he/she can associate this concept when seeing a VW Beatle on the streets.



a) b)

Figure 2 – Triangle of meaning: a) General representation; b) Illustrative example

Examples of symbols used in engineering are drawings, written or spoken words in natural language, alphabetic and numeric characters, computational words (bits, bytes), etc. Therefore, when these symbols stand for an existing object, they carry information.

Technical instruments, such as transducers, signal conditioners, controllers, and analog filters also process information, however, the manipulation of symbols is not immediately evident, but the manipulation of signals. **Signals are observable physical variables, whose state or parameters of variation with time correspond to the symbols that carry information.** The instantaneous value of an analog variable (amplitude) and the pulse duration in a PMW (Pulse Modulated Width) signal are examples of symbols related to the signals. For example, an electrical signal resulting from the measurement of pressure by a pressure transducer carry, through the voltage amplitude (symbol), information about the pressure that is occurring in the measured system and, therefore, it is necessary to establish a correlation between the voltage amplitude and the pressure.

Therefore, it can be noticed that physical systems (hardware) process signals whose symbols are normally not similar to the referent, while in software, symbols are processed directly and they can be chosen in the most convenient way to represent the object.

2.1.3. Characterization of Energy and Material

The **energy** associated with a state of a physical object (system) is a measure of its ability to produce changes in its own states or in the states of other physical objects (systems). The variation of energy in a system results from the transfer of heat and/or work observable in its physical ports, that is, through the interconnection of channels.

Energy ($E(t)$) and, in particular, power ($P(t)$) as the amount of energy transferred or converted per unit time, are completely determined by the generic quantities of effort and flow. That in specific energy domains they assume common designations like force, speed, pressure, voltage, current, etc. (Karnopp, Margolis, & Rosenberg, 1990). Table 1 shows the variables that can be measured in energy ports of a system, whose product results in power. Energy is expressed as

$$E(t) = \int P(t)\, dt. \tag{1}$$

Table 1 – Variables, power and energy in different domains

Domain	Effort	Flow	Power [W]
Generic	$e(t)$	$f(t)$	$P(t) = e(t)\, f(t)$
Electrical	Voltage [V], $U(t)$	Current [A], $i(t)$	$P(t) = U(t)\, i(t)$
Hydraulics & Pneumatics	Pressure [Pa], $p(t)$	Volumetric flow rate [m^3/s], $q_V(t)$	$P(t) = p(t)\, q_V(t)$
Mechanical translational	Force [N], $F(t)$	Velocity [m/s], $v(t)$	$P(t) = F(t)\, v(t)$
Mechanical rotational	Torque [Nm], $T(t)$	Angular velocity [rad/s], $\omega(t)$	$P(t) = T(t)\, \omega(t)$

In addition, **matter or material** is characterized by its chemical, electrical, magnetic, metallurgical, and many other properties. These properties are a consequence not only of the basic substances of which they are composed but also of the type of process used in their manufacture. Examples include shape, dimensions, roughness, conductivity, ductility, etc., as well as thermodynamic properties, such as temperature, pressure, density, mass, and volume.

The **variables characterizing energy or power** and **properties characterizing material** can be designed in general as attributes. The term attribute is commonly used in object-oriented software development corresponding to data stored inside a class or instance, which means, they store information about the instance.

Therefore, the **state of either energy or material resource** corresponds to the values, qualitative or quantitative, assumed by its attributes at a given moment. Likewise, an information resource, which may include one or more symbols, has its status defined by the value assumed by these symbols.

2.1.4. Characterization of Signals

As mentioned before, from a measurement perspective, signals are observable physical variables whose state or parameters of variation with time carry information. From a mathematical point of view, signals are characterized by a dependent variable related to an independent variable through a mathematical function. An independent variable is considered continuous if it assumes all values in an interval of existence, and is considered discrete if it assumes only certain values in an interval of existence. Likewise, dependent variables are classified as continuous and discrete.

Therefore, signals can be classified as follows:

- **Continuous signals (continuous in time):** the independent variable, time, is continuous (Figure 3).
  - **Analog signals:** are continuous signals in time with continuous amplitude;
  - **Quantized signals:** are continuous signals in time with discrete amplitude.
- **Discrete signals (discrete in time):** the independent variable, time, does not assume all values (Figure 4).
  - **Sampled signals:** are discrete signals with continuous amplitude;
  - **Digital or numerical signals:** are discrete signals with discrete amplitude;
    - Binary signals: are digital signals with two discrete amplitude values.

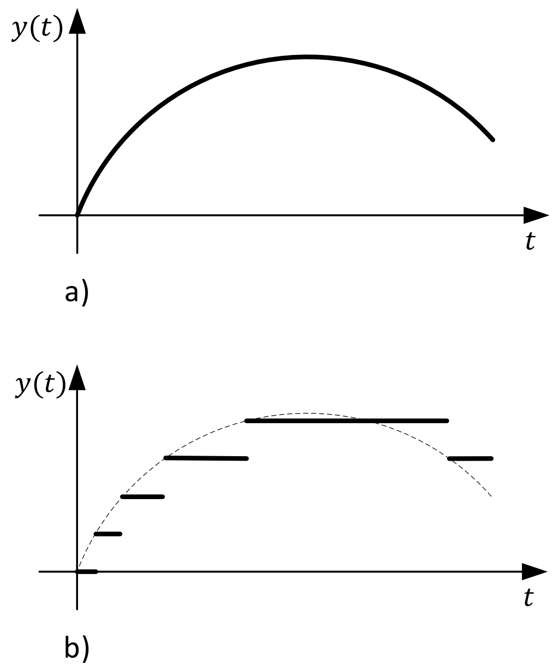


Figure 3 – Continuous signals: a) Analog; b) Quantized

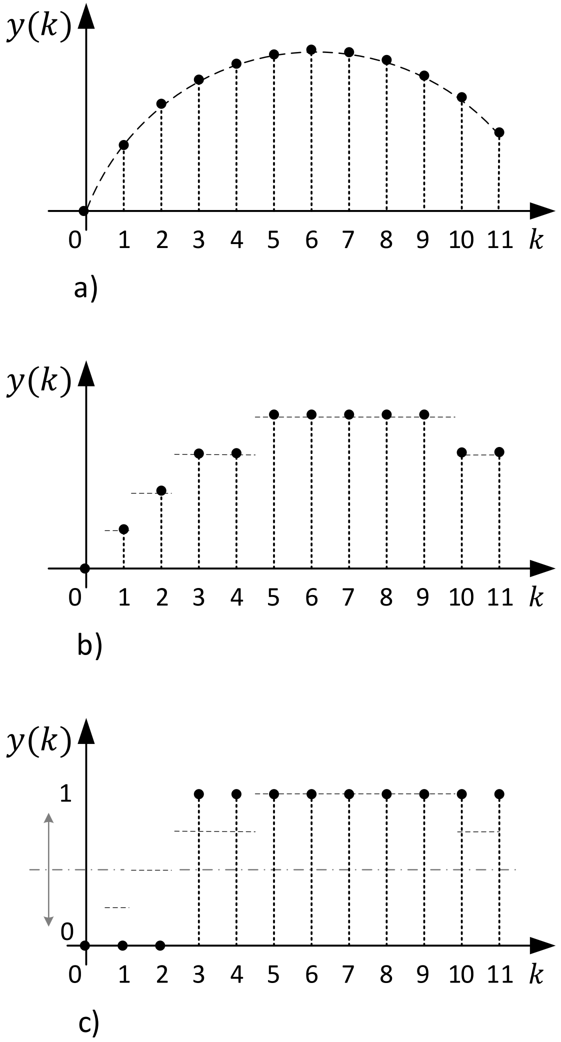


Figure 4 – Discrete signals: a) Sampled; b) Digital; c) Binary

2.2. ABSTRACTION AND MODELING

2.2.1. Introduction

Systems, especially those studied in areas such as mechatronics and software engineering, can usually be divided into several interrelated subsystems. These, in turn, have their own subsystems and so on until the components considered elementary are reached.

This type of structure comprises interactions that are difficult to predict, requiring the study of isolated parts in order to predict the global behavior of the system. It characterizes the systems as complexes, since, as stated by Simon (1981), “given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole.”

As a way of manipulating complexity and facilitating the analysis or design of systems, abstractions can be used. An ***abstraction*** can be seen as a result of applying one or more perspectives to the system or part of it so that the theory provided by the perspectives is used to create ***models*** of those parts of the system (Hoover, Rinderle, & Finger, 1991; Yourdon, 1989).

2.2.2. Model classification according to a perspective

As mentioned above, **models are used to describe systems and are the result of applying one or more perspectives on the systems**. Three main perspectives, structural, functional, and behavioral, have been used to accomplish technical system description of complex systems (De Negri, 1996, 1998) as can be seen in Kumara, Ham, Al-Hamando, and Goodnow (1989) in designing technical systems, as well as in Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen (1991) for software development. Therefore, **modeling of mechatronic or automatic systems essentially must follow structural, functional, and behavioral perspectives.** The junction of these three perspectives is imperative to achieve an essential description of systems and also information integration among the system and other design tools is facilitated.

The **functional perspective** specifies *what the system does or should do.* It describes the function of each component in a system and the relationship among them to achieve a global function. Since objects are usually referred according to the function they perform (Kumara *et al*., 1989), functionality is the main perspective for system specification. As stated by Hubka and Eder (1988), function "refers to something more stable, namely a desire (Buur, 1990) (e.g. to be able to exert a set of effects). This desired ability may not be ensured by the actual behavior of the system, a system can also behave wrongly or badly." Examples of functional models that graphically represent the functions (activities) performed by the system and their correlations are: a) Data flow diagram (DFD) (Yourdon, 1989), b) functional structure used for product design (Pahl & Beitz, 2013), c) the Channel/Agency Net (C/A net) (De Negri, 1996; Heuser, 1990; Reisig, 1985), and d) electric, hydraulic, and pneumatic circuit diagrams (ISO, 2012a, 2012b).

The **structural perspective** describes *where the functions are implemented*. It refers to the set of elements in the system and the set of relationships that interconnect these elements (Hubka & Eder, 1988). These relationships can indicate physical or communication connections or hierarchical relationships to assist possible conceptual associations that can be established between components. The structural model represents the final design objective, which is to establish the system components (hardware and software). The entity/relationship diagrams (Song, Evans, & Park, 1995) and the 2D or 3D technical drawings are some examples of structural models.

Looking through a **behavioral perspective** one can explain *how and when the functions are executed.* Kumara *et al.* (1989) and Henson, Juster, and de Pennington (1994) define system behavior as the relationship of the input from the external environment with the internal state, and the output, or influence, that the system performs over the external environment. Behavioral models have dynamic semantics, that is, they describe changes in both state and output that occur over time.

Behavioral models can be classified according to the type of signal processed by the system (Cassandras & Lafortune, 2008), as presented below:

- **Continuous-state models:** describe the system using state variables, inputs and outputs with continuous amplitude. In these systems, the state generally changes when time changes.
  - **Continuous-time models:** in addition to the amplitude of the signals being continuous, the independent variable is also continuous, i.e., the system operates on analog inputs and produces analog outputs and states; Mathematically, differential equations can be used to describe the relationship between variables, such as:

$$\tau \frac{dy(t)}{dt} + y(t) = K_{ss}u(t) \quad (2)$$

where its solution for a step input of amplitude u_s is

$$y(t) = K_{ss}u_s(1 - e^{-t/\tau}) \quad (3)$$

Systems modeled in this way are called **continuous-time systems**;
  - **Discrete-time models:** the amplitude of the signals is continuous, but the independent variable is discrete, i.e., the input, output and state variables are modified only at discrete instants. Systems modeled in this way are called **discrete-time systems** and are normally expressed through difference equations, such as:

$$y[k] = \left(\frac{\tau}{\tau + T}\right) y[k-1] + \left(\frac{K_{ss}T}{\tau + T}\right) u[k] \quad (4)$$

In both continuous- and discrete-time models, the state variables change continuously over time. For this reason, such systems are called time-driven systems. It is important to note that the time (t in continuous time and k in discrete time) is a natural independent variable, on which all inputs, outputs, and states depend. In addition to the equations presented above, other examples of models are transfer functions, state-space representations, as well as, time response graphs as shown in Figure 5.

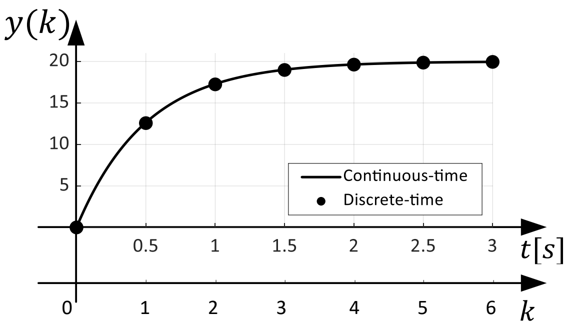


Figure 5 – Time response for a step input

- **Discrete-state models:** describe the system using state variables, inputs and outputs with discrete amplitude. The change from one state of the system to another is called a state transition and occurs only when an instantaneous event occurs. Some examples of events are: pressing a button, detecting that a temperature has been reached in the process or a spontaneous occurrence dictated by nature (e.g. the beginning of nightfall).

Typically, mechatronic systems include a digital controller that operates based on an internal clock. Events may or may not be synchronized with this clock, such that systems can be classified as:

  - **Time-driven (discrete-state) systems:** are systems in which state changes are synchronized with time, since, at each instant marked by the clock, an event (or none) is selected, causing the state transition. The clock, alone, is responsible for any possible state change;
  - **Event-driven (discrete-state) systems:** events occur independently of the instants marked by the clock. Therefore, they are not synchronized with time and not necessarily synchronized with each other (they are asynchronous).

Some examples of event-driven, discrete-state models are Place/Transition Petri Net (Reisig, 1985), automaton (Cassandras & Lafortune, 2008), Boolean logic, and GRAFCET (IEC, 2013).

2.2.3. Model classification according to a representation

Functional, structural, and behavioral models must be expressed in some way in order to be understood by the designer or analyst. The most typical model representations can be summarized as (De Negri, 1996; De Negri & Santos, 2007):

Iconic model: It is a representation, corporeal or not, with a high degree of similarity with the real system. It has a geometric equivalence to preserve the proportions and shapes of the system to be represented. Examples are maps, photographs, plans, models, drawings, etc.;

Analog model: It consists of a relatively easy manipulation system that has a correspondence, normally behavioral, with the system under study or with its variables. There is often little visual similarity between the model and the real system. A typical example is a physical model for tests in a wind tunnel.

Symbolic or mathematical model: It uses logic and mathematics to represent, in an abstract way, the physical laws that are believed to govern the behavior of the system under investigation. It uses idealized elements that have the essential characteristics of the components of the system and have their relationship described through a mathematical expression. Examples of this type of model are mathematical expressions in general, descriptions by state variables and by transfer functions, Boolean equations, etc.

Diagrammatic or schematic model: Composed of a set of lines and graphic symbols that represent structural, behavioral, or functional characteristics of the real system. There is usually little visual correspondence having the ability to describe essential aspects of the system by focusing on a single facet. Examples of this type of model are hydraulic, pneumatic, and electrical circuit diagrams, Entity-Relationship diagrams (ERD), organization charts, bar graphs, X-Y and X-t graphs, state transition diagrams, decision tables, etc.

Natural language model: Description using natural language to represent system characteristics. For example, a list of requirements.

Table 2 presents several models that are applied to the description of system in the scope of mechanical, electrical and software engineering fields

Table 2 – Models classified by perspectives and representations

Model	Perspective	Representation
Functional structure	Functional	Diagrammatic
Block diagram	Behavioral	Diagrammatic and Mathematical
State transition diagram	Behavioral	Diagrammatic
Place/Transition Petri Net	Behavioral	Diagrammatic
GRAFCET	Behavioral	Diagrammatic
Transfer function	Behavioral	Mathematical
Bond graphs	Structural and Behavioral	Diagrammatic
Channel/Agency Petri Net	Functional and Structural	Diagrammatic
DFD – Data Flow Diagram	Functional	Diagrammatic
Electrical circuit diagram	Functional and Structural	Diagrammatic
Hydraulic & pneumatic circuit diagrams	Functional and Structural	Diagrammatic
Entity-Relationship diagram (ERD)	Structural	Diagrammatic
Geometric drawing	Structural	Iconic
State-space equations	Behavioral	Mathematical
Mockup	Structural	Iconic

2.3. CHANNEL/AGENCY NET

2.3.1. Graphical and Mathematical Representation

The **Channel/Agency Net (C/A Net)** is composed of two basic elements: active units represented by rectangles and passive units represented by circles. These elements are connected by directed arcs representing the resource flow (Figure 6). The application of that for automatic/mechatronic system modeling and design was introduced by De Negri (1996) where both structural and functional perspectives can be attributed to the C/A Net while the behavioral aspect is not considered.

From the **functional perspective**, the passive units correspond to the resources that flow throughout the system, that is, the energy, matter, and information or its manifestation forms such as electricity, workpieces, tools, compressed air, signals, data, and so on. The active units, in turn, are activities corresponding to the operations applied to the resources, such as pumping, assembly, transportation, and processing.

From a **structural perspective**, the passive units are channels, indicating those system components that give support to the resource flows without causing modification in their state. Pipes, shafts, wires, magazines, and memories are examples of channels. The rectangles represent the agencies, which correspond to the places where the activities take place, such as pumps, machine components, workstations, chemical reactors, objects (in software), and so on. From this perspective, the C/A net makes the physical link among components or machines explicit.

The graphical symbols and the interpretations according to the two perspectives are shown in Figure 6a. Arrows of the directed arcs are represented according to one of the main types of resources (Belan, 2007; Belan, Szpak, & De Negri, 2010).

It is important to point out that the direction indicated by the arcs that connect the elements in the C/A Net has no meaning under the structural point of view. In this case, they represent the existing interconnection, that is, the way in which the system is constituted. Therefore, it can be concluded that the arcs only indicate which passive component is necessary to establish the connection among the active components. Otherwise, as a functional model, the arrows indicate the resource flow direction.

Therefore, the C/A Net models the interdependence between machines, devices, data processors, and so on, emphasizing the channel through which the energy, matter, or information flows. It is important to emphasize that this notation is not linked to any specific technical area.

Therefore, it can be applied always that functional and/or structural descriptions are necessary.

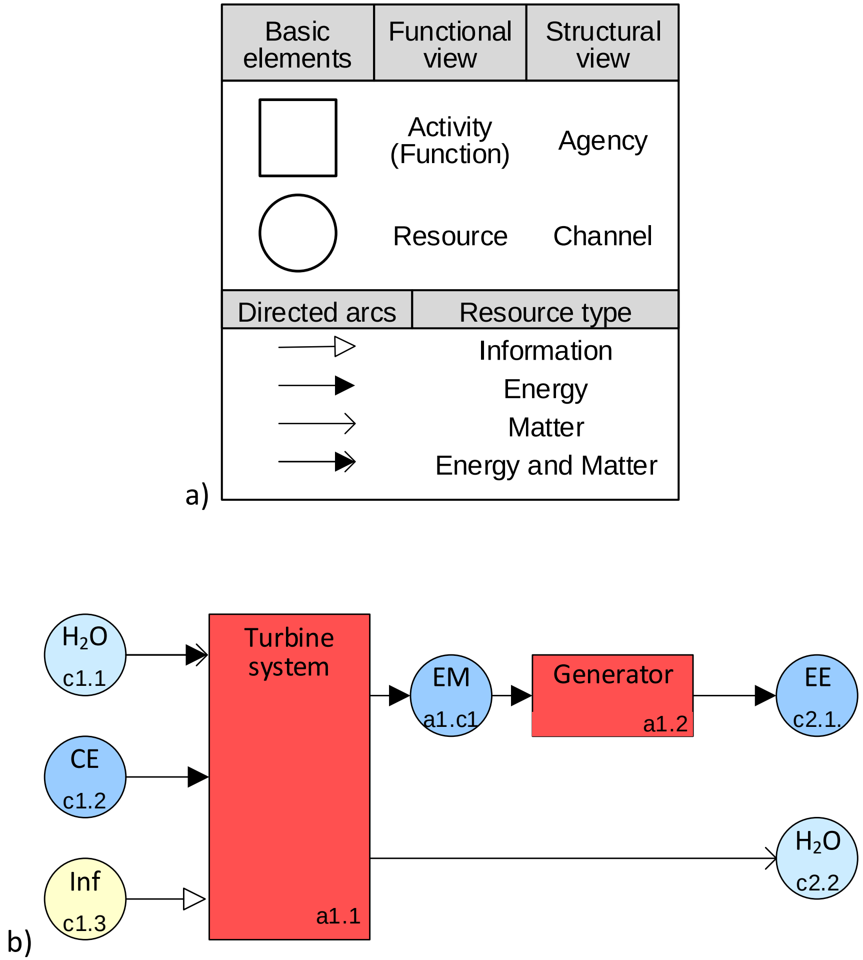


Figure 6 – C/A net: a) Main elements; b) Example of representation

The basic rule for the use of this notation is that the interconnection only is allowed between channels and agencies, that is, two channels or two instances connect by the same arc cannot occur (Figure 6b). **An arc connecting a channel to an agency implies that the activity may depend on the channel content, but not necessarily does**, or, in other words, the resources may be used by the activity. In turn, **an arc that connects an agency to a channel indicates that the channel content may be modified, but not necessarily is**, by the activity, that is, the resource may be modified by the activity (De Negri, 1996; Hanisch, 1992; Heuser, 1990).

As proposed by Belan (2007) and Belan *et al.* (2010), in the same way as ordinary Petri Nets, a formal C/A Net can be mathematically defined by a nonuple. The graphical and mathematical forms differ only in the mode of representation, being very simple to convert a graphical C/A Net to its mathematical form and vice versa.

The main elements of the mathematical description are the pre-incidence matrix (K_{pre}) and the post-incidence matrix (K_{post}). Where K_{pre} is the matrix that defines the input channels to the agencies, taking into account the resources that flow in those channels (preceding channel or previous incidence channel) and K_{post} is the matrix that defines the output channels of an agency, taking into account the resources that flow in those channels (following channel or posterior incidence channel). In both matrices, the lines are the channels (c_i) and the columns the agencies (a_j).

Figure 7 shows the matrix representation of the C/A net presented in Figure 6b. Each matrix element is filled by a bit (zero or one). When the bit is equal to zero it means that there is no arc connecting that channel to that agency. Otherwise, when the bit is one it means that the channel and the agency are connected. Detailed formalization of the mathematical C/A net is found in Belan (2007), where analysis methods checking the structural and resource flow coherences of the C/A Net are presented.

a)

Kpre	a1.1	a1.2
c1.1	1	0
c1.2	1	0
c1.3	1	0
a1.c1	0	1
c2.1	0	0
c2.2	0	0

b)

Kpost	a1.1	a1.2
c1.1	0	0
c1.2	0	0
c1.3	0	0
a1.c1	1	0
c2.1	0	1
c2.2	1	0

Figure 7 – C/A Net: matrix representation: a) Pre-incidence matrix; b) Post-incidence matrix

As will be seen in the next chapters, the C/A Net is used to model the design process or mechatronic systems as well as for documentation of the product under design.

2.3.2. Refinement and Condensation Rules

The refinement of a channel or an agency consists of detailing these, identifying new internal channels and agencies, as illustrated in Figure 8. Likewise, channels and agencies can be grouped together to form condensed elements.

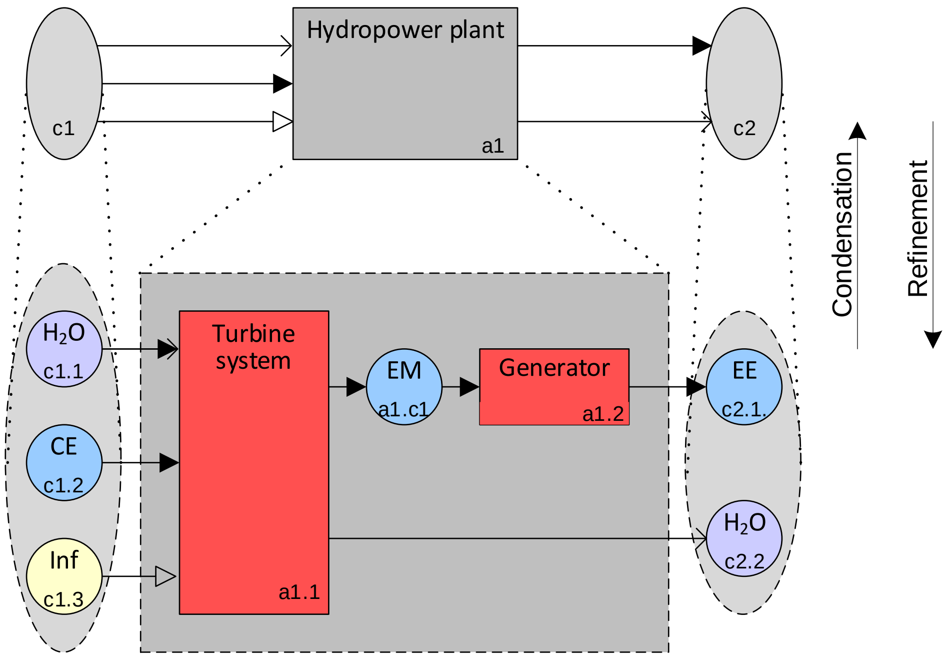


Figure 8 – Refinement and condensation of C/A Net

Both the refined and condensed networks are C/A Nets, so the basic rule of only interconnecting channels and agencies must always be obeyed. When refining a channel, the elements of the refined network that have arcs to outside the channel should also be channels. Likewise, the frontier elements identified in the refinement of an agency must be agencies. Finally, as shown in Figure 8, the arcs in the condensed network must represent all directions of the arcs in the refined network.

3. MECHATRONIC AND AUTOMATIC SYSTEMS

3.1. INTRODUCTION

Technical systems with movement, governing and/or operation capabilities with at least a minimum level of autonomy can be classified as **automatic systems**. Mostly all devices, pieces of equipment, and machines designed nowadays are included in this category. Additionally, they are built using components from different technologies, including sensing, actuation, and information processing. Therefore, current automatic systems are, under the constructive point of view, **mechatronic systems**.

Mechatronic systems are known as those that fundamentally comprise a combination of mechanical, electro-electronic, and information technologies. Mechanical systems include mechanical elements, precision mechanics, micro-mechanical components, and fluid power systems (hydraulics and pneumatics); Electro-electronics comprises electrical machines and components, microelectronics, sensors, and power electronics; and Information technology is a branch that involves system theory, control and automation, software engineering, and artificial intelligence (Buur, 1990; Isermann, 2007)

Mechatronic technology is found in several application fields and machines such as robotics, telecommunications, integrated manufacturing systems, CNC machines, machine vision, automotive engineering (active suspension, automatic transmission, anti-lock braking systems), aeronautical engineering (flight control system, air-conditioning, and heating), autofocus cameras, hard disks, medical imaging systems, household appliances (dishwashers, refrigerators, washing machines), building/home automation, and so on (Wikipedia, 2021).

The design of mechatronic products would be simplified if the system could be separated from the beginning according to the technology (mechanical, electrical, and information) to be used. Consequently, well-established design methodologies could be applied for each area and the final design documentation would be obtained by adding the different design documents. Unfortunately, this approach is not feasible, as it is part of the mechatronic design process to identify the best solution to meet a specific function from a set of technologies. After the conceptual design phase, mechanical, electrical, and information designs will be carried out in parallel, when the interfaces between the technical solutions must be checked continuously.

Therefore, a general and integrative view of a mechatronic system is necessary as well as a mechatronic design methodology with well-integrated phases, steps, and tasks.

3.2. GENERAL ARCHITECTURE OF A MECHATRONIC SYSTEM

Mechatronic systems, as a class of the technical systems discussed in Section 2.1, perform their functions by manipulating and controlling the flow of material, energy, and information resources. Looking at the definition and examples of mechatronic systems presented before, it can be seen that the main function and/or auxiliary functions are performed by producing or changing energy and/or material in some physical ports. It can be producing a mechanical movement, turning on a light, producing a magnetic field, energizing an electrical network, producing a finished piece, blowing air, controlling the air temperature, delivering a product, etc.

As presented in De Negri (1996), a mechatronic/automatic system comprises energy and/or material subsystems that receive energy and/or material from the external environment, process them, and deliver to the environment to accomplish the intended function or functions. The operation of these subsystems depends on the action of an information subsystem, human or not, which is able to extract information from energy and/or material, process, and later on, use to modify the energetic and/or material flows. An external user or other machine communicates with the mechatronic system by information ports. In this way, a mechatronic or automatic system can be modeled as an information subsystem that is attached to an energetic and/or material subsystem by internal information channels. Besides the exchange of information between these two subsystems, there is also the input and output of energy, matter, and information with the external environment.

This architecture is described in Figure 9 using the C/A Net notation (De Negri, 1996; De Negri & Santos, 2007; Santos, Cury, & De Negri, 2001) and it can be interpreted according to two perspectives: From the functional perspective, that means, answering the question “*what do subsystems do?*”, the rectangles represent the activities, identified as Information processing (or to process information) and energy/material processing (or to process energy/material). The circles correspond to the resources, in this case, energy, material, or information; From the structural perspective, answering the question “*where are the functions implemented?*”, the rectangles represent the agencies, identified as information subsystem and energy/material subsystem and the circles are the channels, which means where the resources flow.

As discussed before, the **functional point of view** shows that the activities depend on the resources in the input channels, but does not describe when these resources are used or when the functions are executed. In the same way, the model state that the function changes the resources at the output channels but it does not inform when it occurs. A C/A Net does not include a behavioral perspective, that means, **it cannot be interpreted** as a flowchart representing a workflow or process or a block diagram used in control system theory.

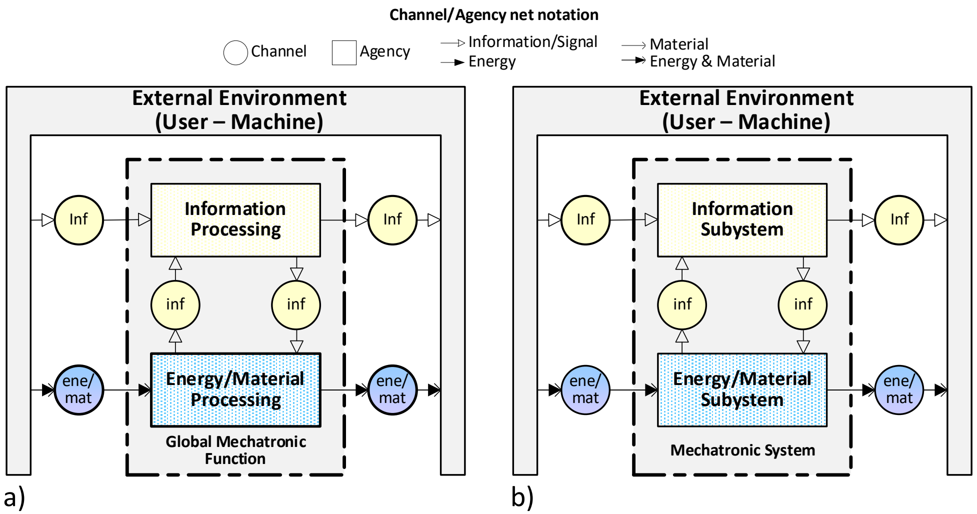


Figure 9 – General architecture of a mechatronic system (condensed view): a) Functional perspective; b) Structural perspective

From the **structural point of view**, the material/energy system is an abstraction of the machines, devices, processes, etc., able to make physical or chemical transformations. The information system involves instruments, programs, human beings, or any other means that process information.

The information flow between energy/material and information subsystems occurs through the measuring and actuation systems. A measuring system (MS) comprises the transducer and/or sensor, signal conditioning unit, and interface. The energy or material is in contact with the transducer or sensor, which results in an information-carrying signal. This signal is amplified and/ or filtered by the conditioning unit and takes the form to be interfaced with a human user or another signal processing system (De Negri, 1996, 1998; Santos et al., 2001). Therefore, the MS is responsible to extract information regarding a specific attribute of the energy or material (see Section 2.1.3).

The actuation system (AS) performs the opposite function, that is, it receives an information signal related to the attribute to be adjusted/controlled, processes it, and, in the end, acts on the flow of energy or material. An AS comprises the interface, signal conditioning unit, and actuator (De Negri, 1996, 1998; Santos et al., 2001).

Figure 10 shows a refinement of the architecture of a mechatronic system where the MS and AS, which are interfacing the (pure) energy/material system (E/MS) with the (pure) information system (IS), are identified. The energy/material system comprises only energy and/or material flows, i.e., does not send or receive information and, consequently, it is not controlled by the information system. Examples of this kind of system are an oil filter installed in a circuit and a gravity roller conveyor, those activities do not require any kind of start/stop command or control signal.

Otherwise, there will be no flow of energy/material through the information system. Its purpose is to process information that may be in the form of a signal. A programmable logic controller, computer, electronic platforms, or even a human operator are examples of systems that perform this function.

Figure 10 also presents the input interface (InI) and output interface (OuI), responsible for communication with other machines or users. Some examples of input interfaces are an electrical switch, touch screen, keyboard, and fieldbus interface module. Output interfaces can be an indicator light, computer screen, siren, and fieldbus interface module. For the operation of electronic instruments processing information, low power energy is usually required. It is represented by the resource auxiliary energy supplied from the external environment.

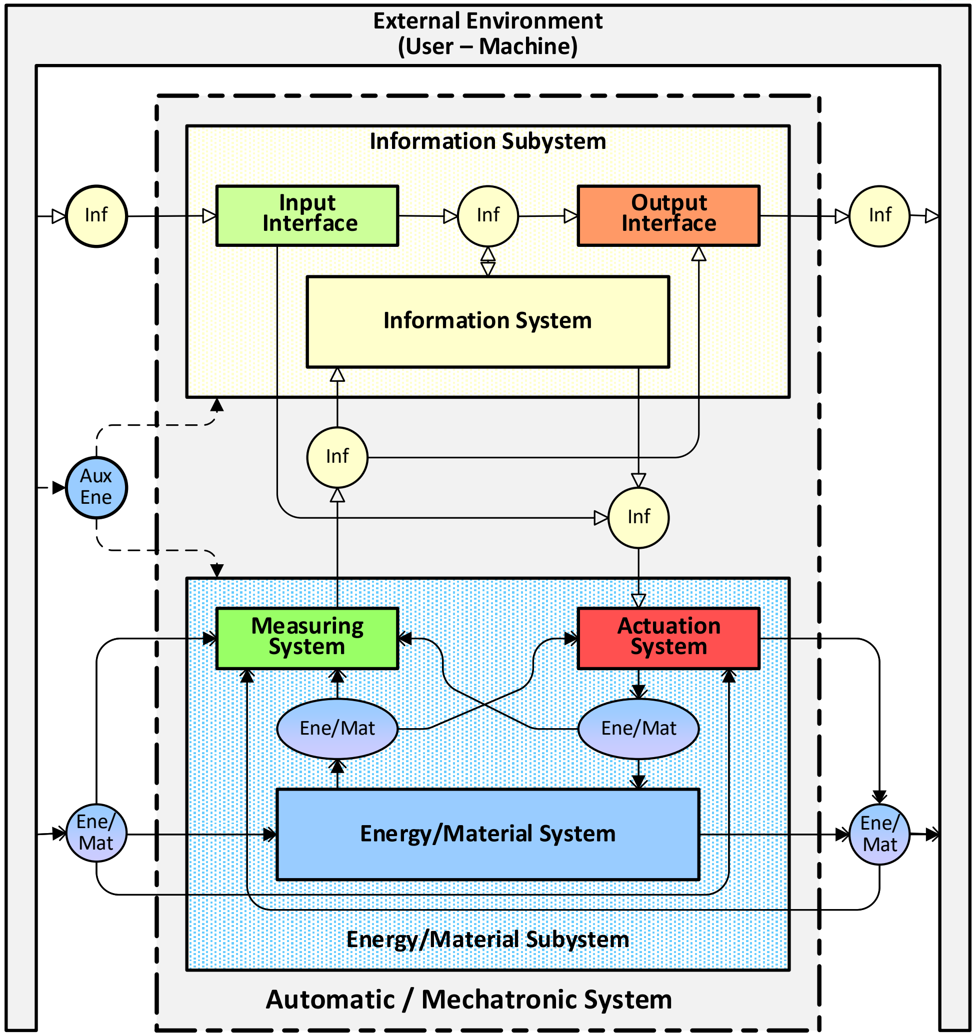


Figure 10 – General architecture of a mechatronic system (refined view)

Therefore, a **mechatronic system architecture comprises six fundamental subsystems**: Energy/Material System (E/MS), Information System (IS), Measuring System (MS), Actuation System (AS), Input Interface (InI), and Output Interface (OuI). Despite Figure 11 showing just one subsystem of each type, one must consider that actual mechatronic equipment or device usually will present several components/subsystems performing the function of E/MS, IS, MS, and so on.

The general principles behind this general architecture are the same presented in Isermann (2007) and VDI (2004). Those authors refer to actuators (or actors) and sensors instead of actuation and measuring systems, and the information system is referred to as a functional block named information processing. The communication with a user and/or machine is in some way indicated too. Moreover, the architecture presented here introduces all resource flows as they will occur in a real system and, therefore, can be interpreted under a structural perspective. It allows to refine progressively this architecture including subsystems (AS, MS, E/MS, IS, InI, and OuI) as necessary as well as to be refining them until the component level.

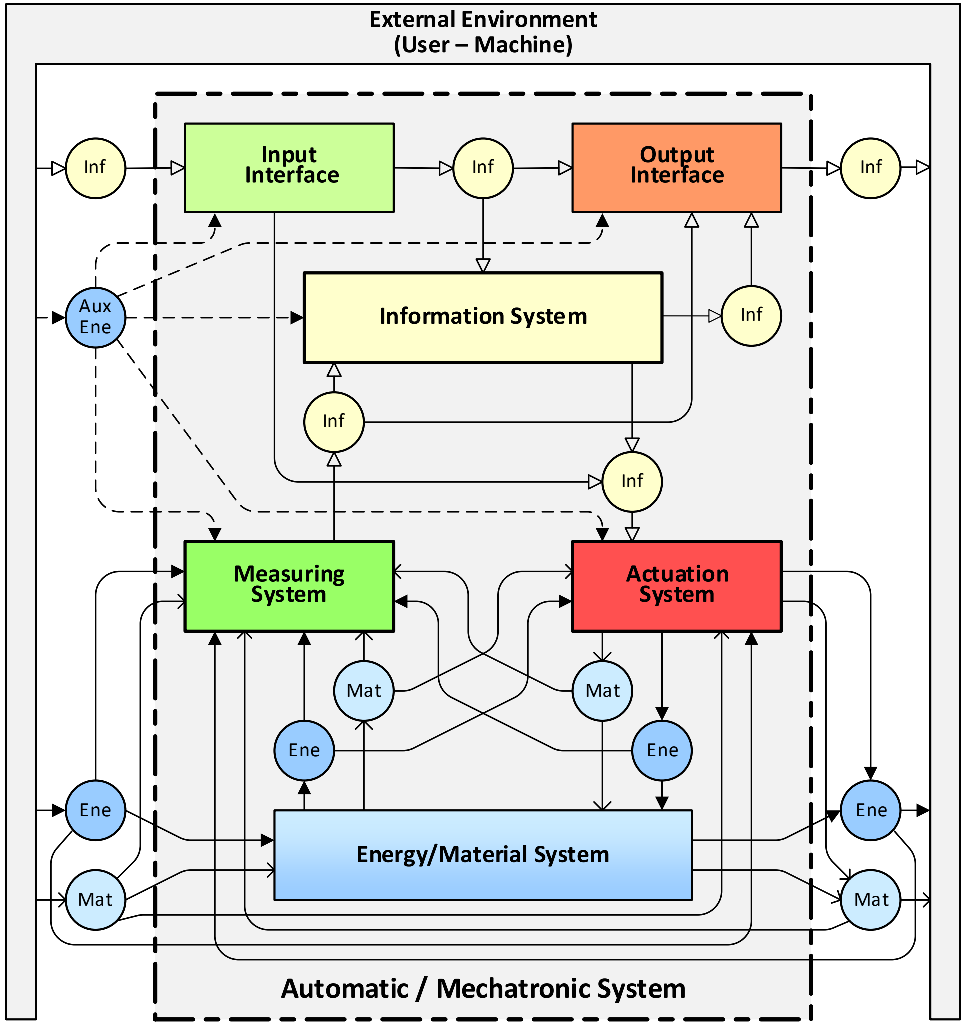


Figure 11 – General architecture of a mechatronic system (refined view, with Ene and Mat separately)

Figure 11 presents a refinement of Figure 10 where the channels of energy and material are shown separately. The energy and material at the input ports can flow through the measuring system (MS), (pure) energy/material system (E/MS), and Actuation system (AS). The MS can also measure variables associated with energy or material flowing from the E/MS and the AS and results on information to the information system (IS). The AS receives information from the IS and controls the flow of material or energy from the external environment or the E/MS. The input interface (InI) receives information from the external environment and after conditioning, it sends either to the information system (IS), the output interface (OuI) and/or the AS. The OuI also receives information from the IS and the MS, resulting in information sent to an external machine or user.

4. DESIGN OF MECHATRONIC SYSTEMS

4.1. INTRODUCTION

In the previous sections it was shown that the modeling of mechatronic systems must contemplate three main perspectives: function, behavior, and structure. The functional perspective, through the C/A Net, is used to represent the general architecture of a mechatronic system and more concrete behavioral and structural models, such as block diagrams, flowcharts, technical drawings, circuit diagrams, etc. allow the detailing of the system parts.

The models mentioned above, together with others applicable to the various technical areas that comprise a mechatronic system, are important tools for the design, analysis, or even for a simple system demonstration. The use of these models becomes evident through the **design process seen as a set of activities that, based on user's requirements, seek to achieve detailed descriptions for the construction or implementation of a technical system.**

The design activities of a mechatronic system occur in different technological domains and, depending on the size of the system, there will be the involvement of several specialists, each capable of solving technical aspects in their area of knowledge. Several publications have deal with the mechatronic design problem. The approach presented herein is based on previous works of Buur (1990), VDI (2004), De Negri and Santos (2007), and De Negri (1996). Moreover, the design phase division as well as several methods and concepts presented in Pahl and Beitz (2013), VDI (1987), and Back, Ogliari, Dias, and Silva (2008) are considered.

As will be presented in the following sessions, some design tasks can be supported by tools implemented in Microsoft Excel. These tools as well as editable diagrams to represent the general architecture of the mechatronic systems are available for download at laship.ufsc.br

4.2. THE DESIGN PROCESS

As will be presented in this section, the design process to be applied for mechatronic/automatic systems is divided in four phases, following the guidelines published in 1977 by Pahl & Beitz (and present in Pahl and Beitz (2013)).

Figure 12 represents the design process using C/A net where the circular forms represent data, information, models, descriptions, etc., i. e., all resources that are necessary to be provided to execute the design phases as well as result from the phase execution. The rectangles represent the design phases, i.e., the general activities that must be carried out in order to reach the product design. As it can be seen, the design process comprises four phases (Pahl & Beitz, 2013; VDI, 1987, 1997, 1998):

- Planning and task clarification
- Conceptual design
- Embodiment design
- Detailed design

The drawing style presented here is similar to the classic waterfall representation (Jacobson, Christerson, Jonsson, & Overgaard, 1994; Jalote, 2005), giving the idea that the design preferably occurs in sequential phases, where each phase depends on the results of the previous one. However, the arcs of the outcomes from design phases that return to the previous phases indicate that the design task is interactive, and the decisions made can be revised as needed. Observe also that the design specifications include data that are used in all phases.

Furthermore, there are arcs in both directions between a phase and the corresponding outcome. It represents that during the phase execution the outcome is being created and, at the same time, it is reused to be revising the phase steps and tasks and, consequently, to update the outcome. As an example, consider the planning and task clarification phase which results in a list of design requirements. As the list is created, the designer reads it and collects additional information about the problem, revising his/her decisions and updating the list.

The design input is the problem/need that can be observed in the environment in which the user or organization is inserted and from where the information can be obtained to carry out the design planning and task clarification. The first result from the design activity is the design data and specifications, comprising the first representation of the system under design. The next result is a conceptual model that formalizes the requirements, mainly functional, and establishes one or more generic structures capable of solving the problem. Up to this point, the system under design is not yet feasible. The definitive model incorporates more refined specifications and changes in the components and structure in order to achieve a feasible design, making it possible to carry out behavioral simulations and build hardware and software prototypes. As a result of the last phase of the design, the product model includes the components list, software final tests, manufacturing process planning, circuit assembly instructions, etc.

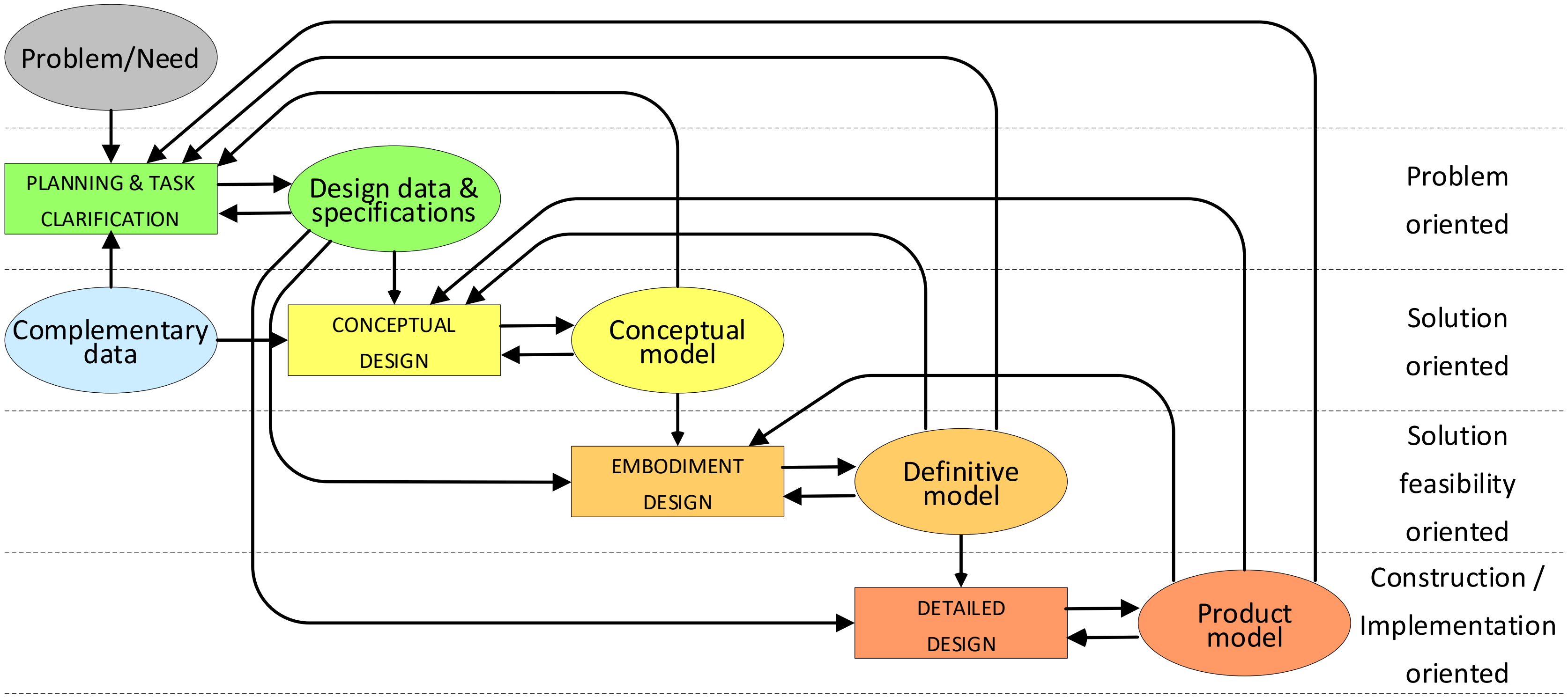


Figure 12 – Design process model

As mentioned before, a mechatronic system comprises several parts, each one designed or built from a dominant technology. Not just electrical, mechanical, and information technologies are present but others like optoelectronics, vision, laser, etc. are applied during the design process requiring the participation of design teams with different backgrounds. Therefore, the design flow shown in Figure 12 starts considering the whole system but as subsystems are identified, parallel design activities will take place.

The result of each phase expresses the current state of the system under design. For its representation, it is usually necessary to use a set of models, each representing a functional, structural or behavioral perspective. Of course, it is desirable to use models known to specialists in each technical area, such as circuit diagrams, technical drawings, data flow diagrams (DFD), etc.

4.3. PLANNING AND TASK CLARIFICATION

4.3.1. Introduction

As shown in Figure 12, planning and task clarification requires the collection of user data, including people and/or organizations, where the problem or need may be latent and will then be identified in this design phase. The information source can be as diverse as possible, which can be obtained from:

- Client/user needs;
- Stimuli from market, environment, own company, etc.;
- Products of the company;
- Products in literature and patents;
- Competitors’ products;
- Standards, guidelines, regulations;
- Technical and scientific literature.

As presented in Pahl and Beitz (2013), the stimuli for designing a new product comes from the external and internal sources: Market stimuli include the position of the company’s products in the market, changes in the market requirements, requests from customers, etc.; Environment stimuli are related to economic and political changes, emerging technologies, environmental protection rules, safety issues, etc.; and company’s stimuli can be new ideas and results from company’s research, identification of new functions, the introduction of new production methods, and so on.

Once an opportunity to develop a new product or improve an existing one is perceived, current products, technologies, documents, and diverse literatures consist of an important source of information to support product design.

Figure 13 represents the **planning and task clarification phase**. It is a functional model using C/A Net notation where the circular shapes correspond to data, documents, reports, etc. that are the inputs and outputs of the steps and tasks to be carried out during the phase execution. The Place/Transition Petri Net (referred to as the Petri Net for short in the rest of this book) presented on the right side is used to represent the sequence of task execution, starting with the initial task (denoted by the unconnected transition/arc pointing to Circle 1) and ending with the final task (denoted by the transition/arc pointing out of Circle 5). The transitions (horizontal lines between the circles) indicate the completion of the preceding task. As it can be seen, this phase comprises **two main steps, the product planning and the task clarification**, and they are divided in tasks as will be discussed below.

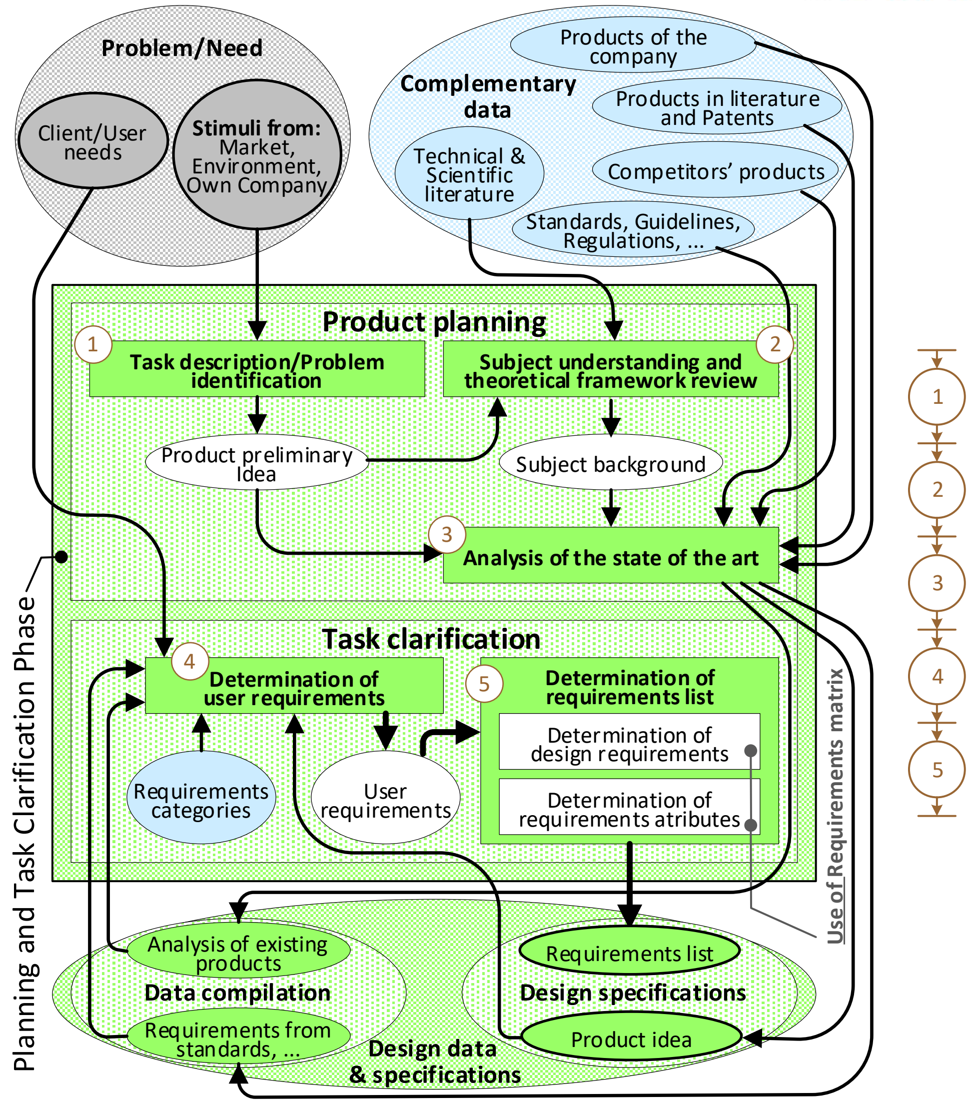


Figure 13 – Planning and task clarification phase

4.3.2. Product Planning

The first step, the product planning, comprises three main tasks:

- Task description/Problem identification;
- Subject understanding and reviewing of theoretical framework;
- Analysis of the state of the art.

The decision to be starting a product design is inspired by the company's external or internal factors as mentioned before. The description/problem identification task, therefore, formulates the preliminary idea of the product, basically a written proposition in which the motivation and the general objective are documented.

The next task consists of research, read, and analysis of technical and scientific literature to get a full understanding about the subject associated to the new product. It includes learning about the application field of the product and the physical and engineering fundamentals.

With the formulated product preliminary idea and the subject background, it is possible to search for current products in the own company, produced by competitors, patented or published in books, technical magazines, scientific journals, etc. In the same way, it is recommended to be looking for technologies that can be applied for the product development, as well as to check the legal constraints and restrictions published in national and international standards. Manufacturing, assembly, reusability, etc. guidelines applied for similar products and legal regulations stated by government, industry associations and national and international organizations must be analyzed as well. This task is named Analysis of the state of the art in Figure 13 and the results can be grouped in three main segments:

- Analysis/comparison between existing products and technologies;
- Report about requirements from standards, guidelines, regulations, etc.;
- Formulation of the product idea/product concept.

As presented in Back *et al.* (2008), the product idea, or product concept, consists of a set of technical and commercial characteristics. Technical issues include a functional description, main structural characteristics, product attributes under the user point of view, etc. Target price, production volume, and market segment are some of the commercial attributes that must be associated with the product idea as well. The innovative characteristics, product life expectation, and technology level must be also linked to the product idea.

4.3.3. Task Clarification

The second step in this design phase is the task clarification, which is divided in two tasks:

- Determination of user requirements (customer requirements);
- Determination of requirements list, including the following subtasks:
  - Determination of design requirements (target requirements);
  - Determination of the requirements attributes including demands and wishes, absolute and relative weights (importance), technical and/or economic impact, as well as, the values associated to the design requirements and the applicable verification methods.

While the user requirements are typically expressed in the language of the customer, the design requirements will be written using an engineering/technical language.

4.3.3.1. Determination of User Requirements

Based on the results achieved in the product planning step shown in the ellipses at the output of the analysis of state of art, it is possible to determine a list of requirements demanded by the user/client. In order to make it easier to identify the requirements, it is useful to identify categories of attributes/characteristics that are expected for the new product, that means, the categories of the user requirements. Table 3 shows a suggested classification for categories and some examples to help the designer to write the user requirements according to each specific product design. The categories are based on Back *et al.* (2008) and Pahl and Beitz (2013).

The user requirements have a natural and compact language, usually composed by generic requirements such as: work correctly, easy to maintain, low cost, nice looking, easy to produce, easy to transport, etc.

Table 3 – Categories and examples of user requirements

Requirements categories	Examples of user requirements	Requirements categories	Examples of user requirements
Functionality	Remove dirty	Maintainability	Easy/quick maintenance
	Harvest cocoa		No skilled labor required
	Wood chipping	Packability	Safe packing
	Lift weight		Compact packing
	Move weight	Transportability	Be portable
Ergonomics	Low vibration	Usability	Compact dimensions
	Low noise		Easy of transportation
	Ergonomic handling		Easy operation
Aesthetics	Nice look	Recyclability	Intuitive operation
	Simple finish		Recyclable materials
	Robust appearance		Recyclable residues
Safety	Low risk of accidents	Geometry	Simple geometry components
	No movable parts exposed		Suitable dimensions
Reliability	Reliable operation	Kinematics	Smooth movements
Legality	Comply with security laws		Low impact
Patentability	Innovative design	Force	Don't damage interfacing objects
Standardization	Comply with applicable standards	Energy	Renewable energy
Robustness	Be durable	Materials	High efficiency
	Be robust		Non-conductive material
Environmental impact	Environment friendly	Automation	Organic material
	Do not pollute		Autonomous operation
Manufacturing	Easy manufacturing	Time	Fast operation
	Low cost manufacturing	Costs	Minimal acquisition cost
Assemblability	Suitable component arrangement		Economical operation
	Few components		Low cost materials
	Easy assembly		

4.3.3.2. Determination of Requirements List

In order **to transform the user requirements into a requirements list**, the proposed requirements matrix shown in Figure 14 is used in the present design methodology. The **requirements matrix** is based on the house of quality, which is a resulting chart from the Quality Function Deployment (QFD) methodology (Ullman, 2016). However, it uses a reduced quantity of subjective and qualitative data in order to achieve a confident requirements list. The index $j = 1\ to\ n$ corresponds to the user requirements and the index $i = 1\ to\ k$ to the design requirements.

The first input of the requirement matrix is to list what are the user requirements. To plan the desired quality of the product an **Importance Factor** (IF_j) for each user requirement is used, where it is expected a qualitative assessment from the user with the following possible values:

1 – Less significant;
2 – Significant;
3 – More significant;
4 – Mandatory.

The following step is the conversion of user requirements into engineering characteristics, which are known as design requirements and represent attributes of the product that can be manipulated, measured, and have a target value (Back *et al*., 2008; Pahl & Beitz, 2013; Ullman, 2016). In essence, the design requirements answer the question of *how* the user requirements can be achieved. It is important to note that all the design requirements have to be assigned to a measurable value (*How much*), with which a verification method can be used to ensure that the user requirements are being met.

An example of a requirements matrix is presented in Figure 15 related to the design of a washing machine. The user requirement of “Fast operation” can be accomplished by the design requirements “Agitation” and “Load capacity”.

Every design requirement will have a technical and/or economic impact on the design of the new product. For instance, the use of a material resistant to corrosion will have a significant impact on the manufacturing budget, therefore it is classified as an economical requirement. On the other side, the requirement “Agitation” is classified as a technical requirement due to the challenges that it imposes during the design. Also, some requirements present either technical and economic challenges, in these cases, they have both technical and economical classification (T/E).

Importance factor (*IF*)

1 - Less significant
2 - Significant
3 - More significant
4 - Mandatory

$$\begin{aligned} &if\ IF_j = 4\ and\ R_{i,j} = 2 \\ &\qquad then \rightarrow D \\ &else \qquad\quad \rightarrow W \end{aligned}$$

$$W_{a_i} = \sum_{j=1}^{n} R_{i,j} \,.\, IF_j$$

$$W_{r_i} = \left\lfloor \frac{4.\,W_{a_i}}{max(W_{a_{1\ldots k}})} \right\rceil$$

$$where\ j \in [1, n]\ and\ i \in [1, k]$$

User requirements (What)	**j = 1**	**2**	**3**	**4**	**5**	**6**	**…**	**n**	**Requirements list**						
	User requirement 1	User requirement 2	User requirement 3	User requirement 4	User requirement 5	User requirement 6	…	User requirement n							
IF	1	3	2	1	4	2	…	4	**Design requirement (How)**	**Demand or Wish**	**Absolute weight (W_a)**	**Relative weight (W_r)**	**Technical/ Economic impact**	**Values (How much)**	**Verification Method**
i = 1	0	1	2	0	1	2	…	1	Design requirement A	W	19	**3**	T	Design req. A value	Verification method A
2	1	2	1	1	2	1	…	2	Design requirement B	D	28	**4**	T	Design req. B value	Verification method B
3	2	1	2	1	2	2	…	1	Design requirement C	D	26	**4**	E	Design req. C value	Verification method C
4	2	2	2	0	1	1	…	1	Design requirement D	W	22	**3**	T	Design req. D value	Verification method D
⋮	⋮	⋮	⋮	⋮	⋮	⋮	…	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
k	1	2	1	1	2	1	…	1	Design requirement K	D	24	**3**	T/E	Design req. K value	Verification method K
Status	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**							

Relationship ($R_{i,j}$)
0 - None
1 - Middle
2 - Strong

Each design requirement must have at least one strong (2)

Fill-in fields
Selection fields
Auto fill

Impact
E - Economic
T - Technical
T/E - Both

Figure 14 – Requirements matrix

User requirements (What)	j = 1	2	3	…	n							
	Remove dirt	Fast operation	Be robust	…	Minimal acquisition cost	**Requirements list**						
IF	4	3	4	…	4	**Design requirement (how)**	**Demand or Wish**	**Absolute weight (W_a)**	**Relative weight (W_r)**	**Technical/ Economic impact**	**Values (How much)**	**Verification Method**
i = 1	2	2	1	…	1	Provide Agitation	D	22	**3**	T	30 minutes of agitation	Prototype testing - Measure agitation time
2	0	2	2	…	2	Adequate load capacity	D	22	**3**	T	10 kg of dry clothes	Prototype testing - Measure load weight
3	0	0	2	…	2	Resistant to corrosion	D	16	**2**	T/E	No rust due water contact	Visual inspection
⋮	⋮	⋮	⋮	…	⋮	⋮	….	….	⋮	⋮	⋮	⋮
k	1	2	2	…	2	Low manufacturing cost	D	26	**4**	T/E	40% of selling price	Manufacturing budget
Status	**OK**	**OK**	**OK**	**OK**	**OK**							

Figure 15 – Example of a requirements matrix

Each user requirement needs to be correlated with all the design requirements. The possible **Relationship** (R_{ij}) values are:

0 - No correlation;
1 - Weak correlation;
2 - Strong correlation.

It is expected that every user requirement has at least one strong correlation with a design requirement, meaning that a formal statement (how) of this user requirement has been provided. The correlation analysis is useful to assign numerical weights to the design requirements, providing means to determine their overall importance during the decision-making activities of the design process.

The **Absolute Weight** (W_a) and **Relative Weight** (W_r) for each i design requirement are calculated, respectively, by

$$W_{a_i} = \sum_{j=1}^{n} R_{i,j}.IF_j \tag{5}$$

and

$$W_{r_i} = \left\lfloor \frac{4.W_{a_i}}{max(W_{a_{1\ldots k}})} \right\rfloor, \tag{6}$$

where $j = 1\ to\ n$ is s the order number of the user requirement and $i = 1\ to\ k$ the number of the design requirement. The relative weight will be an integer value between 0 and 4.

The column "Demand or Wish" aims to identify the mandatory design requirements, which need to be attended by the final product. The requirements classified as wishes are derived from user requirements with an importance factor between 1 and 3, meaning that for the user, they are not mandatory. Therefore, if during the design process a mutually exclusive decision needs to be made, the design requirements classified as wishes are the first to be looked at. The classification between demand (D) or wish (W) is made according to

$$\begin{array}{l} if\ IF_j = 4\ and\ R_{i,j} = 2 \\ \qquad then \rightarrow D \\ else \qquad\quad \rightarrow W \end{array} \qquad where\ j \in [1, n]\ and\ i\ \in\ [1, k] \tag{7}$$

As can be seen in the extended column "requirements list" of Figure 14, it comprises an insightful set of engineering specifications that ensures a correct understanding of the problem and that the customer needs will be attended. Moreover, the requirements list along with the product idea and the data compilation performed during the first phase of the design process will be used as the main source of information during the entire design of the new product, helping to select different conceptions, components, materials, and behaviors.

As shown in Figure 13, the result of the planning and task clarification phase are the **design data and specifications** which comprises a **data compilation** resulting from the Product Planning and the **design specifications** from the Task Clarification.

Data compilation includes:

- Analysis of existing products;
- Requirements from standards, guidelines, etc.

Design specifications include:

- Product idea;
- Requirements list.

It is important to emphasize that the design data and specifications can be updated at any time during the design process. As represented in Figure 12, conceptual, definitive, and product models can be used to be reworking this first design phase, resulting in new or reformulated requirements. When either embodiment or detailed design starts, the designer can observe that some design specifications were not included, mainly because the conceptual architecture has defined several subsystems not previously considered.

4.4. CONCEPTUAL DESIGN

4.4.1. Introduction

Conceptual design is the phase in which the designer changes from the point of view of the problem to that of the solution. Starting from the **design data and specifications**, the functional requirements are translated to a functional decomposition and the identification of operational means (working principles) that can execute the selected functions. Since a mechatronic system combines several technologies and, therefore, requires a multidisciplinary approach, the conceptual design must result not only one functional diagram (block diagram) as usual for mechanical system design, but a **conceptual integrated design** and **models in specific domains**. The first one comprises at the same time functional and structural perspectives using the formal representation by Channel/Agency Net (C/A Net). The specific-domain models are mainly graphical representations proper from each technical domain showing functional and/or structural characteristics. A behavioral description must be contemplated at this design phase too but compatible with the technical means identified at this point. Despite of occurring system synthesis at this time, the result is still in an intermediary level of abstraction.

In the present work, the conceptual design is modeled according to

Figure 16, which is divided in two steps:

- Functional decomposition and means synthesis
- Synthesis of general structure and behavior.

The first step aims to identify the technical functions and operational means based on the product idea and it should be capable of fulfilling the list of requirements, as well as the regulatory requirements. Complementary data available in the literature and products in the market must be analyzed in order to support the functional decomposition and identification of candidate solutions. The second step begins with the achieved solution(s) and creates a set of diagrams, equations, and even natural language descriptions constituting the conceptual model. For each solution, a corresponding conceptual model will be derived.

The Petri net shown on the right in Figure 16 represents the sequence in which tasks should be preferably be executed, as discussed in sections 4.4.3 and 4.4.4.

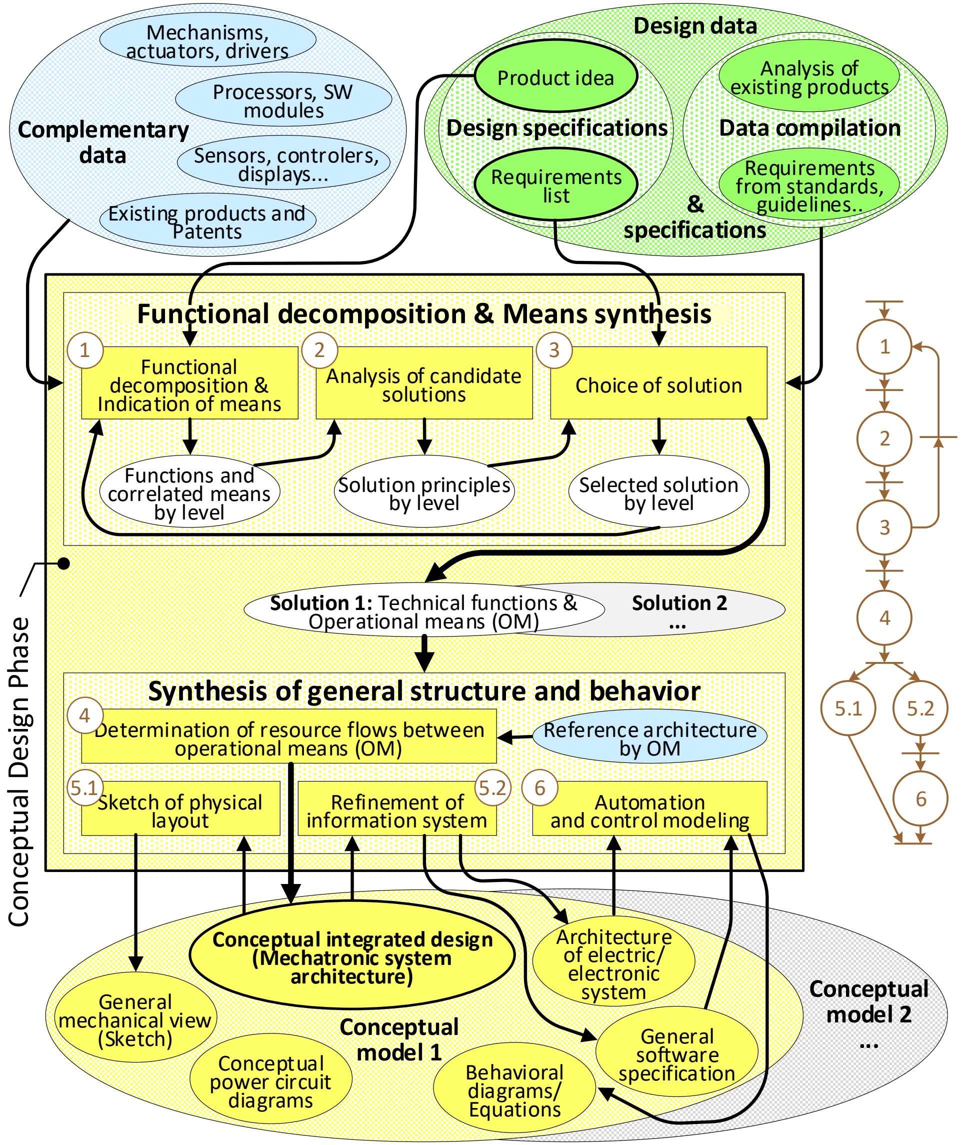


Figure 16 – Conceptual design phase

The conceptual design approach presented herein is a scenery where the approaches applied for the different technical domains are considered. Methodologies for technical systems, mostly constituted by mechanical parts, use functional decomposition to create technical solutions, as presented in Back *et al.* (2008), Pahl and Beitz (2013), and VDI (1987). Otherwise, software development starts with a combination of functional, structural, and behavioral representations of the design requirements, and these three perspectives are used through the detailing of the software (Nygaard, 1986; Object Management Group, 2021; Rumbaugh *et al.*, 1991). Electronic system design is also carried out by behavioral and structural decomposition, with very low emphasis on the functional description (Gajski & Kuhn, 1983; Gerstlauer *et al.*, 2009). Mechatronic design methodologies have not given so much emphasis on functional decomposition as seen in Isermann (2007), VDI (2004), and Buur (1990), but focusing on the integration of domain-specific designs instead.

The present work uses functional decomposition as the starting point of the systems conception, however, combined with synthesis of solution principles in each level of abstraction. For that, the Function-Means tree (F/M Tree) combined with morphological matrices and evaluation charts are used to perform the step functional decomposition and means synthesis shown in Figure 16. These three methods are presented in the next sections and the step functional decomposition is discussed in detail in Section 4.4.3.

4.4.2. Support Methods

4.4.2.1. Function/Means Tree

Function of an object or system is the effect it has on the external environment; the ability to perform a set of actions. **Means** is a general description for a working principle, a technical principle that can realize the required function. Once a function is formulated, it is possible to designate alternative means, which may carry out the desired function (Buur, 1990).

The Function/Means Tree (F/M Tree) is a “hierarchical arrangement of function levels and means levels, connected with lines that correspond to the causal relations between functions and means” (Hansen & Andreasen, 2002). This representation enforces a strict alternation between function and means nodes as the levels of abstraction are lowered.

The functions are represented by trapezoids and means by rectangles. From each function, several means can be identified obeying an OR logic function and one of them will be chosen to move to the next level of abstraction. From the chosen means, more than one subfunction may be identified following an AND logic function. The means in one specific level can be represented separately with an OR logic relation with the previous function or listed in one box. Figure 17 presents an example of a generic F/M Tree using these two different graphical representations.

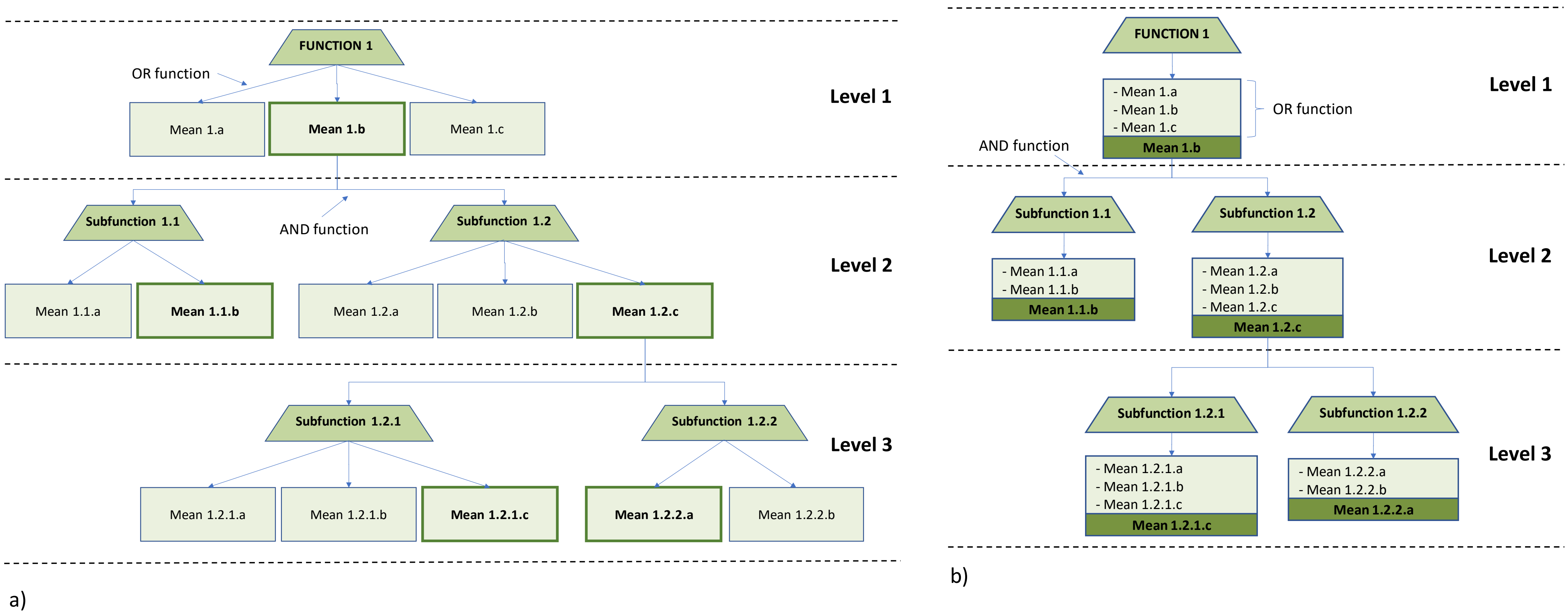


Figure 17 – Example of Function-Means Tree: a) With means represented separately; b) With means grouped in a box.

At Level 1, the functions are presented by one digit. Usually, only one global function is typified, but in some cases, the product to be designed must be described by two or more main functions. For example, two simultaneous global functions could be ***Delimit rooms*** **(Function 1)** and ***Protect against fire*** **(Function 2)** for a Product idea of "a system to be installed in an area of a commercial building with the objective of delimiting the work areas without loss of visual contact and that at the same time fire does not spread through the floor of the building".

At Leve2, the subfunctions are identified by two numbers. The first digit represents the number of the original function and the second one distinguishes it from the other subfunctions at the same level. For example, Level 2 includes Subfunctions 1.1 and 1.2 in Figure 17. The same rule applies to subsequent levels such that the number of digits typifying a function/subfunction is the same as the Level number they belong to.

4.4.2.2. Morphological Matrix

Morphological matrix, or morphological method, is a known tool for creating alternative solutions (solution principles) by the combination of working principles for sub-functions, as those identified previously in the F/M Tree. This technique is used in many design methodologies (Back et *al.*, 2008; Pahl & Beitz, 2013; Ullman, 2016; Ulrich & Eppinger, 2016; VDI, 2004) since it encourages the creation of several different solutions, stimulating the creativity and imagination of the design team.

The method can be summarized in three steps (Pahl & Beitz, 2013; Ullman, 2016):

1) To list the sub-functions that must be accomplished in the first column of a matrix;
2) To find as many working principles as possible that can perform a subfunction. They are listed in the row corresponding to the sub-function.
3) To choose one working principle for every sub-function resulting in a solution principle. Different combinations can be identified resulting on several solution principles.

The resulting solution principles are concepts able to perform the set of subfunctions considered in the morphological matrix. To move to the next design activity, the design team must choose one of these concepts and it will determine a branch of the design process. If necessary, another concept can be evaluated later, and the design activities followed by this other branch.

Considering the modeling perspectives discussed in Section 2.2.2, the working principles and subsequent solution principles are representations under the ***structural point of view***, i.e., they represent *where the functions can be implemented*. Therefore, the morphological matrix is a way to correlate structural models to ***functional models***, which represent *what the system (or subsystem) does or should do*.

Figure 18 – Morphological method: a) Morphological matrix; b)Figure 18a illustrate a morphological matrix comprising n sub-functions (1.1, 1.2 … 1.n). For each sub-function, at least one working principle must be included by a brief text description and preferentially an illustration. The working principles are the means identified in the F/M Tree and they are identified by a sequence of numbers and a letter. The numbers correspond

to the subfunction identification and the letters distinguish the applicable working principles. Selected working principles are interconnected by lines resulting in a solution principle. Figure 18b shows some selected concepts from the morphological matrix.

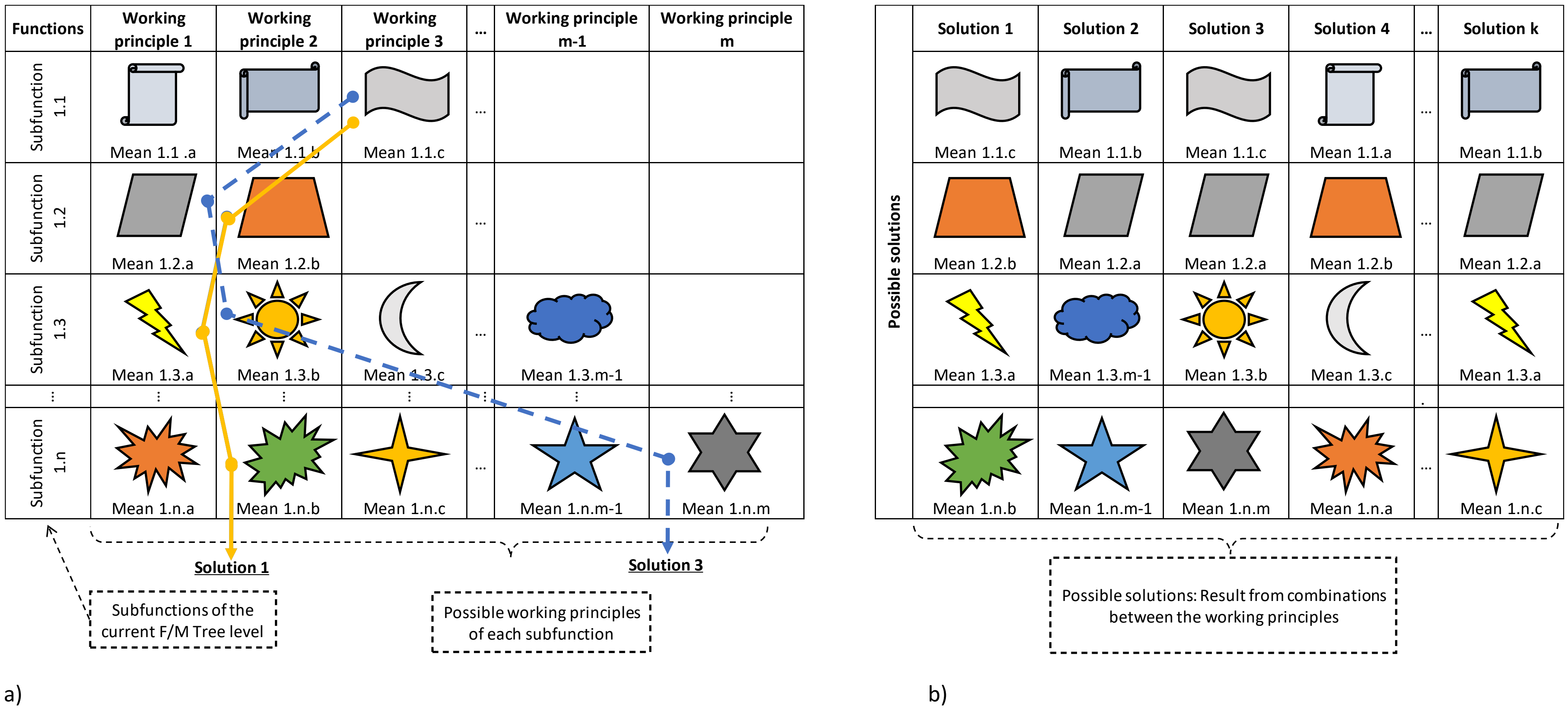


Figure 18 – Morphological method: a) Morphological matrix; b) Synthesis of the selected solution principles

4.4.2.3. Evaluation Chart

For evaluating alternative concepts, as those resulting from the morphological matrix, the *decision-matrix method (Pugh's method)* is presented in literature which provides means of scoring each alternative concept relative to the others in its ability to meet the criteria (Ullman, 2016). Similar to that, VDI (1997), VDI (1998), and Pahl and Beitz (2013) propose the use of an Evaluation Chart that allows a combined technical and economic evaluation. Technical criteria include categories listed in Table 3 such as ergonomics, environmental, safety, and so on.

The evaluation procedure is not restricted to the conceptual design phase, but it can support decisions at the other phases too. Therefore, more abstract solutions can be analyzed as well as concrete systems as devices, components, pieces of software, etc.

In the design methodology presented here, the evaluation char is used for analyzing solution principles identified as promising concepts to execute a set of functions/subfunctions, being the criteria extracted from the design requirements (Figure 14). Additional criteria could be provided by general constraints however it is proposed here that they would be included previously as design requirements.

The evaluation chart is shown in Figure 19 and, as mentioned before, it is composed by two sections, the technical and economic analysis. In each section, the design requirements are allocated according to their classification in the requirements list (Figure 14) and the **Relative Weight** (W_r) is transferred to this chart. Subsequently, a cross analysis is performed between the solution principles and the design requirements, where a **Solution Weight** (W_S) is assigned to describe how well it attends the technical or economic criterion and it can assume the following possible values (VDI, 1997, 1998):

0 – Unsatisfactory;
1 – Just tolerable;
2 – Adequate;
3 – Good;
4 – Very good.

The evaluation chart includes a hypothetical optimal solution used as a reference, which is placed in the column on the right. An **Optimal Solution Weight** (W_S) equal 4 is attributed to all criteria and it will result in the maximum possible total values that a solution may have.

Design criteria are composed of design requirements. (Adopt only design criteria useful for each evaluation chart)

Technical criteria

T and T/E design requirements

Same design criterion might be present in both analysis

Economic criteria

E and T/E design requirements

Relative Weight

Possible means/solutions

Optimal mean/solution

Evaluation chart											
	Design criteria	j=	1		2		3		...	o	
		W_r	**Solution 1**		**Solution 2**		**Solution 3**		**...**	**Optimal**	
			W_{s1}	$W_{s1}.W_r$	W_{s2}	$W_{s2}.W_r$	W_{s3}	$W_{s3}.W_r$	...	W_o	$W_o.W_r$
i =	**Technical analysis**										
1	Design requeriment A	4	4	16	2	8	3	12	...	4	16
2	Design requeriment C	2	2	4	0	0	3	6		4	8
3	Design requeriment F	2	3	6	3	6	4	8	...	4	8
4	General constraint A	3	3	9	3	9	4	12	...	4	12
⋮	⋮	⋮	...	...	...	...	...	...	...	...	...
n	General constraint D	2	3	6	4	8	2	4	...	4	8
	Total technical value		**T_{t1}**	**41**	**T_{t2}**	**31**	**T_{t3}**	**42**	...	**T_{to}**	**52**
	Relative technical value		**R_{t1}**	**0.7885**	**R_{t2}**	**0.5962**	**R_{t3}**	**0.8077**		**R_{to}**	**1**
i =	**Economic analysis**										
m	Design requeriment B	3	4	12	2	6	3	9	...	4	12
m+1	Design requeriment F	2	2	4	3	6	0	0		4	8
m+2	Design requeriment G	4	3	12	3	12	4	16		4	16
m+3	General constraint D	2	4	8	4	8	4	8		4	8
⋮	⋮	⋮	...	...	...	...	...	...	...	...	...
p	General constraint H	1	4	4	4	4	4	4	...	4	4
	Total economic value		**T_{e1}**	**40**	**T_{e2}**	**36**	**T_{e3}**	**37**	...	**T_{eo}**	**48**
	Relative economic value		**R_{e1}**	**0.8333**	**R_{e2}**	**0.75**	**R_{e3}**	**0.7708**		**R_{eo}**	**1**
	Hyperbolic rating		**H_{r1}**	**0.81**	**H_{r2}**	**0.67**	**H_{r3}**	**0.79**	...	**H_{ro}**	**1**

Assign values for each mean/solution (Ws1, ..., Wo) to describe how well it attends the technical or economical criterion

0 - Unsatisfactory

1 - Just tolerable

2 - Adequate

3 - Good

4 - Very good (Ideal)

$$T_{t_j} = \sum_{i=1}^{n} W_{Sj_i}.W_{r_i}$$

$$R_{t_j} = \frac{T_{t_j}}{T_{t_o}}$$

$$T_{e_j} = \sum_{i=m}^{p} W_{Sj_i}.W_{r_i}$$

$$R_{e_j} = \frac{T_{e_j}}{T_{e_o}}$$

$$H_{r_j} = \sqrt{R_{t_j}.R_{e_j}}$$

Figure 19 – Evaluation chart

For each j possible solution, a **Total Technical Value** (T_t) and a **Total Economic Value** (T_e) are calculated, respectively, by

$$T_{t_j} = \sum_{i=1}^{n} W_{S_{j,i}}.W_{r_i} \tag{8}$$

and

$$T_{e_j} = \sum_{i=m}^{p} W_{S_{j,i}}.W_{r_i}. \tag{9}$$

Furthermore, a **Relative Technical Value** (R_t) and a **Relative Economic Value** (R_e) are obtained, respectively, using

$$R_{t_j} = \frac{T_{t_j}}{T_{t_o}} \tag{10}$$

and

$$R_{e_j} = \frac{T_{e_j}}{T_{e_o}} \tag{11}$$

An overall analysis of the possible solutions is made using the **Hyperbolic Rating** (H_r)), which is the square root of the product of the **Relative Technical Value** (R_t)) and **Relative Economic Value** (R_e)) of the respective j possible solution:

$$H_{r_j} = \sqrt{R_{t_j}.R_{e_j}} \tag{12}$$

This quantity helps to balance eventual distortions between the technical and economic aspects, giving a higher rating for the balanced solutions (Andreasen, Hansen, & Cash, 2015; Pahl & Beitz, 2013). The best solution of a specific level will be the one with the highest hyperbolic rating.

4.4.3. Functional Decomposition and Means Synthesis

4.4.3.1. Introduction

As shown in Figure 16, the step Functional Decomposition and Means Synthesis includes three tasks, that are:

- Functional decomposition and indication of means;
- Analysis of candidate solutions;
- Choice of technical solution.

Remembering the understanding of a C/A Net, it is a functional/structural model and, consequently, it does not describe when the activities will be executed. However, there is a causal dependence since an activity (performed by the agency) requires the existence of resources in the input channels, and the execution of an activity results on information at the output channels.

Therefore, analyzing Figure 16, the first task will be the **functional decomposition and indication of means**, resulting in a set of *functions and correlated means*. This result will be the input for the task **analysis of candidate solutions (solution principles)**. Based on the S*olution principles* for the mechatronic product, the **choice of solution** is performed. As mentioned before, the F/M Tree can have several levels such that the three tasks above are executed by level and the final result (*selected solution*) will be used for the functional decomposition on the next lower level. As the decomposition achieves operational means a *technical solution* will be achieved. This sequence of task execution is represented by the Petri net shown on the right in Figure 16.

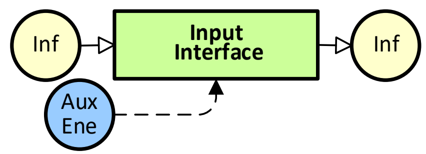
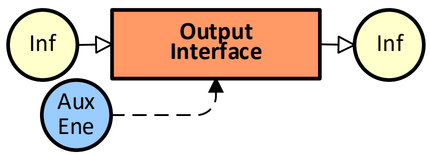
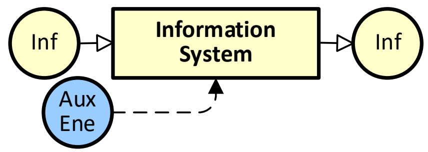
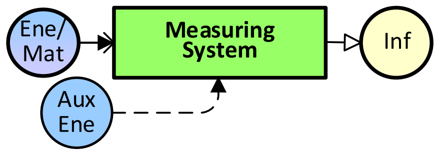
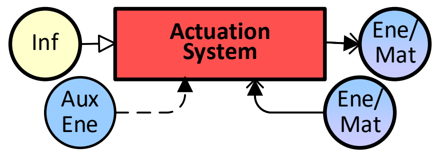
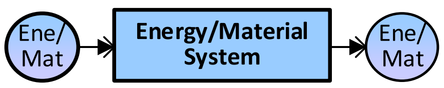
4.4.3.2. Functional decomposition and indication of means (Using F/M Tree)

The construction of a F/M Tree (Figure 20) starts by the definition of a global function, which is a condensate statement that objectively describes what the product should do. After that, the design team must look for abstract means, fundamentally technologies that are capable to accomplish this global function. For example, if the global function is to generate electricity, possible technologies would be solar, hydropower, wind power, thermoelectric, and nuclear. The selection of means can be supported using evaluation chart as will be discussed later.

In order to move to the level 2 of the F/M Tree, the global function must be decomposed in further subfunctions taking in consideration the selected mean. From level 2 to the bottom, for each subfunction, several means can be identified and one of them will be chosen with the support of both morphological matrix and evaluation chart. The progressive decomposition aims to break down the global function into a set of specific subfunctions.

As the subfunctions are decomposed in lower levels, the corresponding means eventually will become physical components or software functions, those which the design team does not want to further decompose. These are called operational means and denote the end of the functional decomposition of the branch. In order to create an integrated view of the main components of the system, the operational means are classified according to their main role, according to the categories presented in Table 4.

Table 4 – Operational means (fundamental mechatronic subsystems)

Operational mean	Symbol	Diagram
Input interface	InI	
Output interface	OuI	
Information system	IS	
Measuring system	MS	
Actuation system	AS	
Energy/material system	E/MS	

It is important to highlight that not all the tree's branches need to finish in the same level, since some branches may represent a more complex subfunction and require further decomposition. Figure 20 shows a generic F/M tree including comments on how to build it. Figure 21 presents an example of a F/M tree of a washing machine.

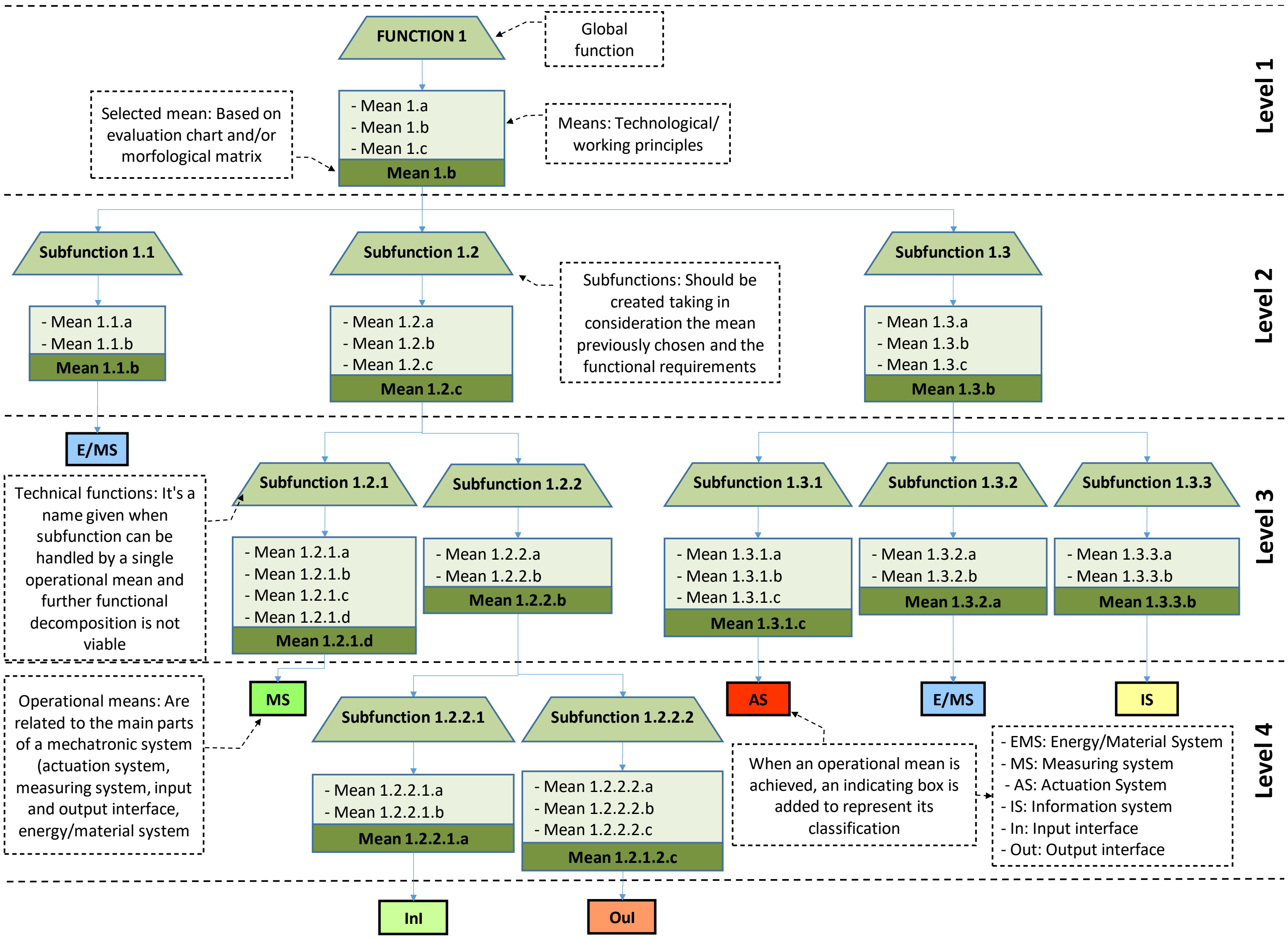


Figure 20 – Function/Means tree for the task Functional decomposition and indication of means

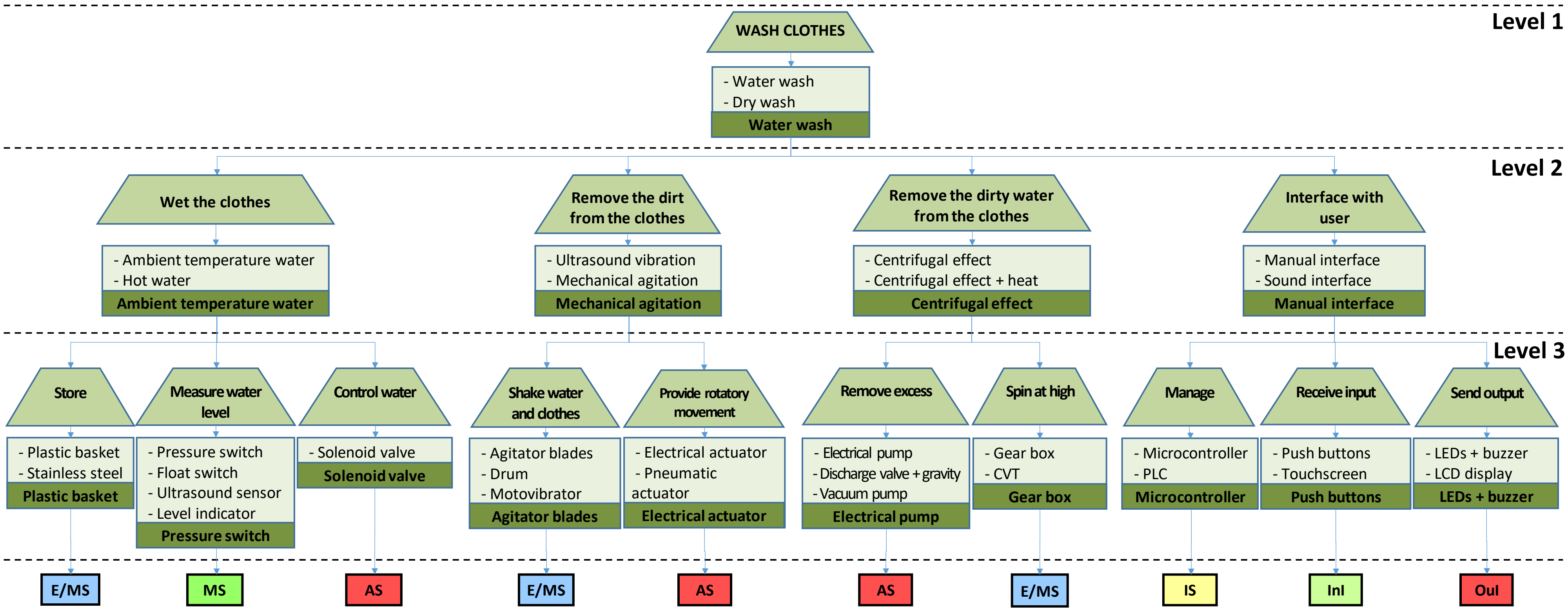


Figure 21 – Function/Means tree of a washing machine

4.4.3.3. Analysis of Candidate Solutions (Using Morphological Matrix)

As discussed above, a set of means suitable to execute a function or subfunction must be identified by the design team. The search of means, or working principles, can be done using data compilation resulted from the first phase of the design – the Planning and task clarification - (Figure 13) as well as complementary data reachable from product catalogs, books, technical reports, research papers, standards, existing product, etc. The search can result in the identification of existing or proposed solutions in several application areas that can be useful to solve the design problem, including:

- Mechanisms, actuators, and drivers;
- Processors, electronic hardware, and software modules;
- Sensors, controllers, and displays;
- Components in finished products.

Methods for supporting the search of working principles and finding new products are described in literature such as Pahl and Beitz (2013), Back *et al.* (2008), Ullman (2016), and Andreasen *et al*. (2015). Among these methods, the morphological matrix discussed previously helps the designer to organize potential working principles applicable to perform functions as well as to combine them to achieve a solution principle.

The morphological matrix is applied from the second level of the F/M tree for describing in natural language and/or by a picture the means reported below to each function in the F/M tree. Figure 22 presents one example of the morphological matrix developed for the second level of the F/M tree of Figure 21.

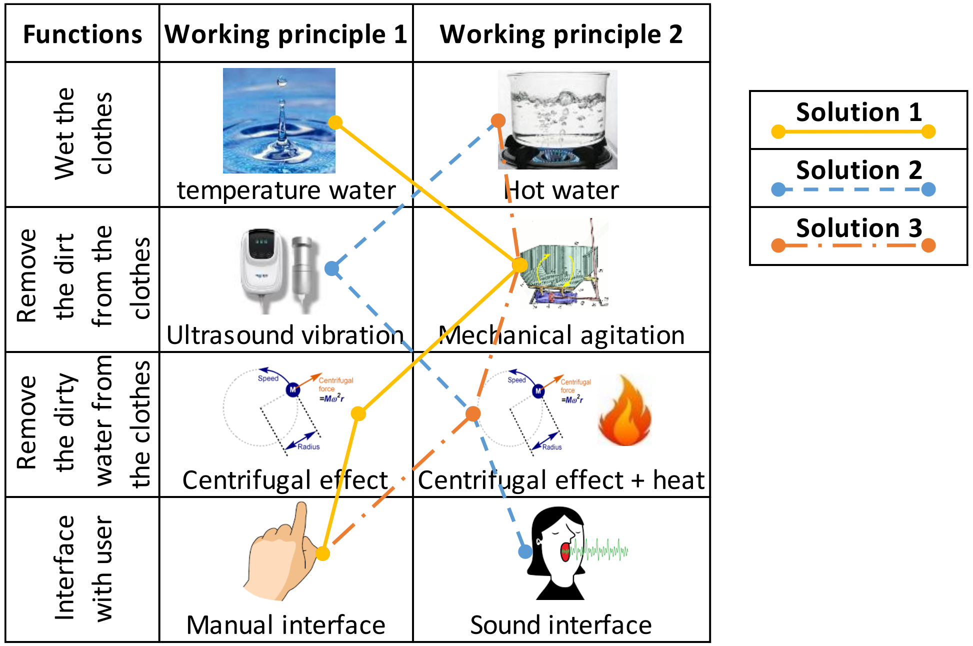


Figure 22 – Morphological matrix from level 2 of the F/M Tree (Figure 21)

4.4.3.4. Choice of Technical Solution (Using Evaluation Chart)

From the morphological matrix of each level of the F/M tree, an evaluation chart is built for supporting the choice of one solution principle, that means, a concept formed by one working principle for each subfunction.

The evaluation chart related to the second level of the F/M tree (of Figure 21.) is shown in Figure 23. As it can be seen, the design requirements and the relative weights (W_r) are linked to the requirements list (Figure 15). It is important to highlight that depending on the functions that are being analyzed and the abstraction level, some design requirements would not be relevant and might be considered as a design criterion. For instance, the requirement “Load capacity” would be relevant on the first level because the dry wash process is complex and cannot handle big loads, however, on the second level, all possible solutions can be designed to have the same load capacity and, therefore, this design requirement does not contribute to the analysis.

Using the evaluation chart, the design team must to analyze and compare the possible solutions with the design requirements. This process stimulates discussions within the team and results in a well understanding of the proposed solutions. Moreover, it ensures that the design requirements are being effectively used for the selection of means.

Evaluation Chart										
	Design Criteria	j=	1		2		3		o	
		W_r	Solution 1		Solution 2		Solution 3		Optimal	
			W_{s1}	$W_{s1}.Wr$	W_{s2}	$W_{s2}.Wr$	W_{s3}	$W_{s3}.Wr$	W_o	$W_o.W_r$
i=	**Technical analysis**									
1	Agitation	3	4	12	3	9	4	12	4	12
2	Low weight	2	3	6	4	8	3	6	4	8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	Resistant to corrosion	2	3	6	4	8	3	6	4	8
	Total technical value		T_{t1}	**24**	T_{t2}	**25**	T_{t3}	**24**	T_{to}	**28**
	Relative technical value		R_{t1}	**0.86**	R_{t2}	**0.89**	R_{t3}	**0.86**	R_{to}	**1.00**
i=	**Economic analysis**									
m	Resistant to corrosion	2	4	8	3	6	4	8	4	8
m+1	Manufacturing cost	4	4	16	3	12	3	12	4	16
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
p	Maintenance cost	2	4	8	3	6	3	6	4	8
	Total economic value		T_{e1}	**32**	T_{e2}	**24**	T_{e3}	**26**	T_{eo}	**32**
	Relative economic value		R_{e1}	**1.00**	R_{e2}	**0.75**	R_{e3}	**0.81**	R_{eo}	**1.00**
	Hyperbolic rating		H_{r1}	**0.93**	H_{r2}	**0.82**	H_{r3}	**0.83**	H_{ro}	**1.00**

Figure 23 – Evaluation chart from level 2 of the F/M Tree (Figure 21)

4.4.4. Synthesis of General Structure and Behavior

4.4.4.1. Introduction

As seen above, the functional decomposition in the F/M tree occurs until operational means are found in all branches of the tree. Therefore, a transformation from a functional perspective to a structural perspective took place, and now an interconnection between subsystems as well as a general behavior of the mechatronic system can be determined.

As shown in Figure 16, the step of **Synthesis of general structure and behavior** includes the following tasks:

- Determination of resource flows between operational means;
- Sketch of physical layout;
- Refinement of information system;
- Automation and control modeling.

The first task is represented by the state 4 in the Petri net on the right side of Figure 16 and it will be followed by the simultaneous execution of tasks 5.2 and 5.2. Once a first version of the Refinement of information system is accomplished, the task 6 can be executed.

4.4.4.2. Determination of resource flows between operational means

As discussed in Section 3.2, mechatronic systems have a common general architecture where each part can be categorized as one of the six fundamental subsystems, or operational means, listed in Table 4.

Figure 10 or Figure 11 is, therefore, the **Reference Architecture by Operational Means (OP)** indicated in Figure 16. If preferable, instead of using C/A Net notation, the system architecture can be drawn as in Figure 24, where circles representing channels are substituted by labels in the arcs. The label can identify the name of the channel or the resource that is flowing inside it.

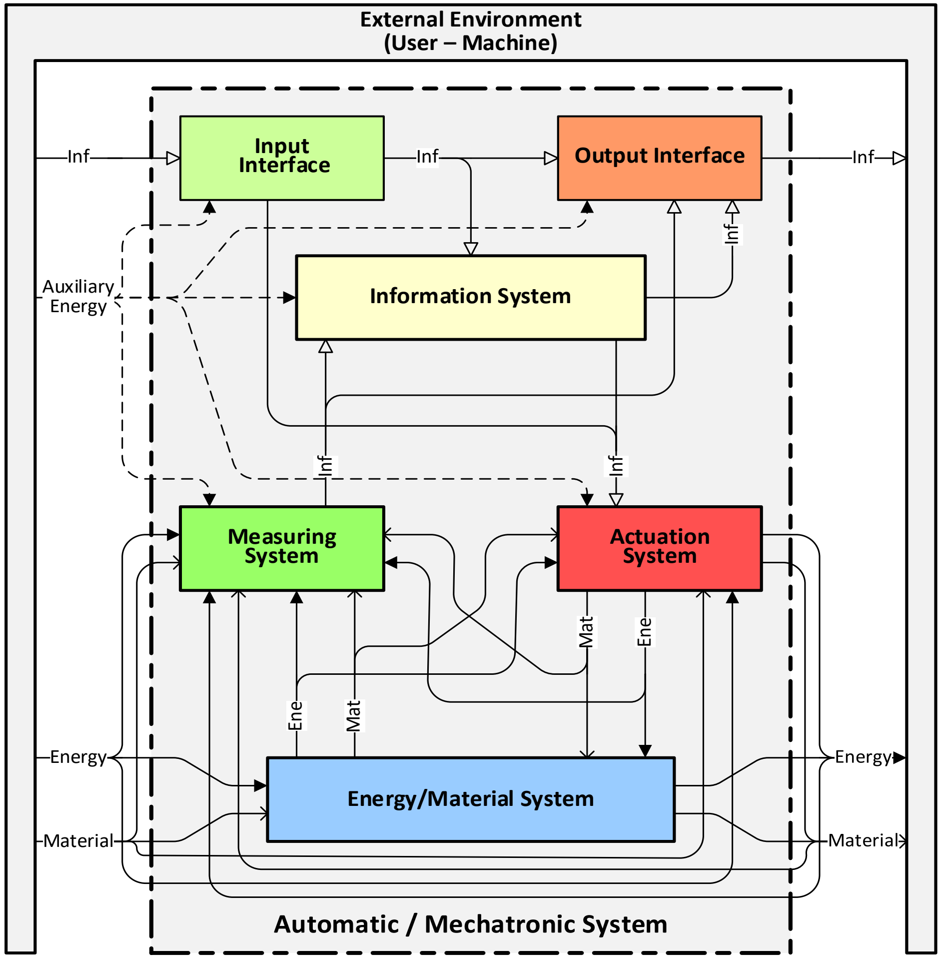


Figure 24 – General architecture of a mechatronic system (*Reference Architecture by Operational Means)*

As mentioned above, once all branches at the F/M Tree have ended on operational means, they should be organized in such a way that it is possible to have an understanding of how the means are connected to each other and how the resources (energy, material, and information) are flowing within the system. Therefore, the result of this task is an integration diagram named "Mechatronic system architecture" that summarizes the conceptual integrated design.

As an example, the corresponding mechatronic system architecture of the washing machine (Figure 21) is presented in Figure 25, where it can be seen the flow of material (water, soap, and clothes), energy (mechanical and electrical), and information.

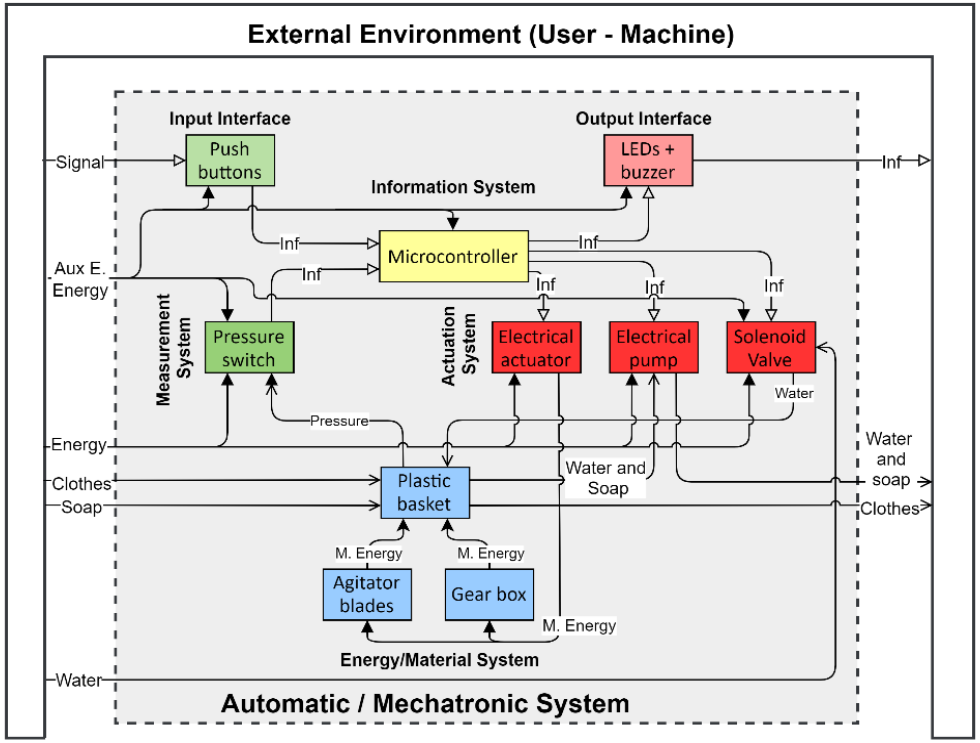


Figure 25 – General architecture of the washing machine

4.4.4.3. Sketch of physical layout

After the mechatronic system architecture has been created, a first draft of the product concept can be developed. This includes mechanical drawings and circuit diagrams, which clarify the role of the selected operational means and detailing the physical connections and spatial arrangement of the system components.

An approximate scale drawing is essential to describe the physical location of the main parts and provide a better understanding of the selected solution. Where electrical, hydraulic and/or pneumatic technologies are used, a preliminary power circuit diagram must be provided, which is the standard representation of actuators, valves, pumps, motors, switches, relays, and so on and their interconnections. Figure 26 presents a sketch of the washing machine and Figure 27 a related water hydraulic circuit diagram.

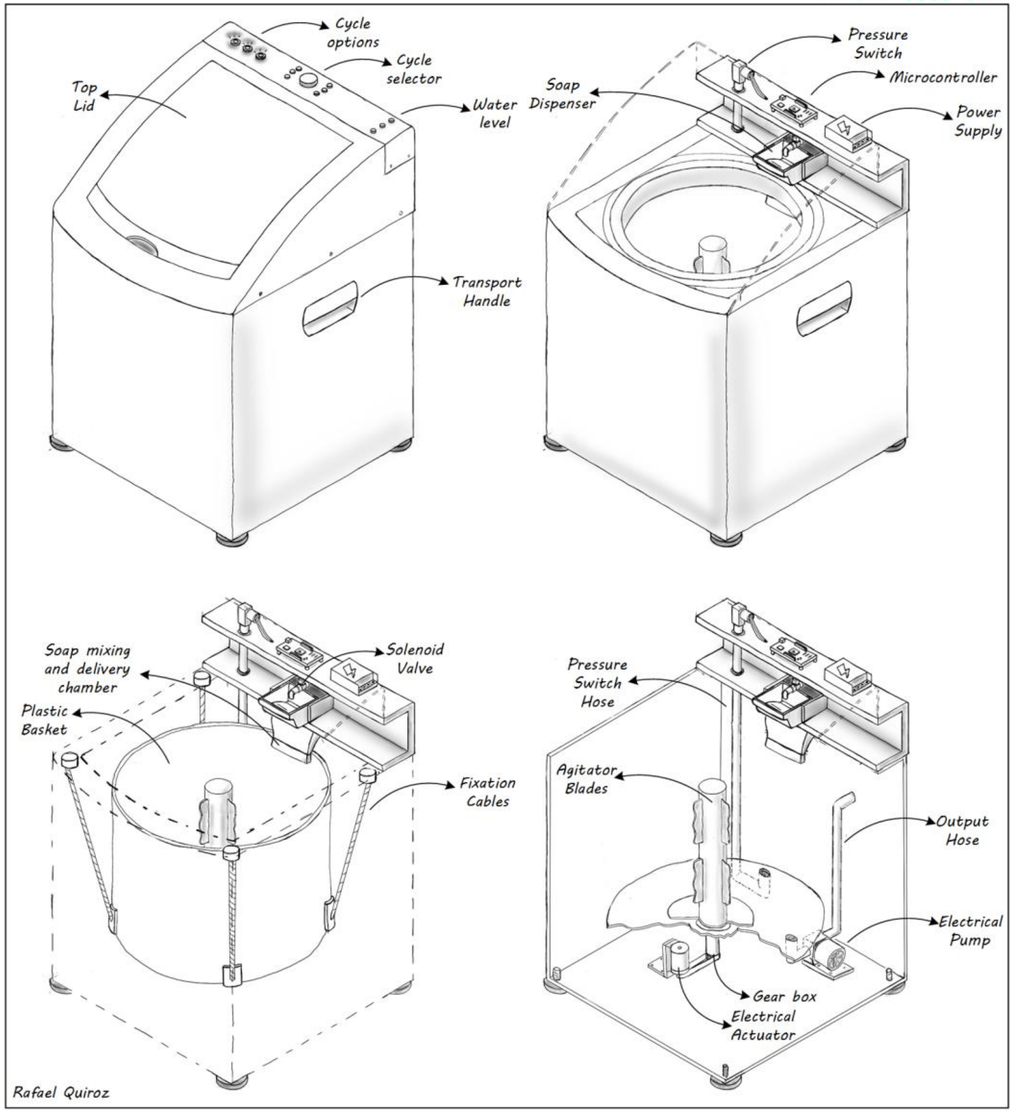


Figure 26 – Sketch of the physical layout of the washing machine

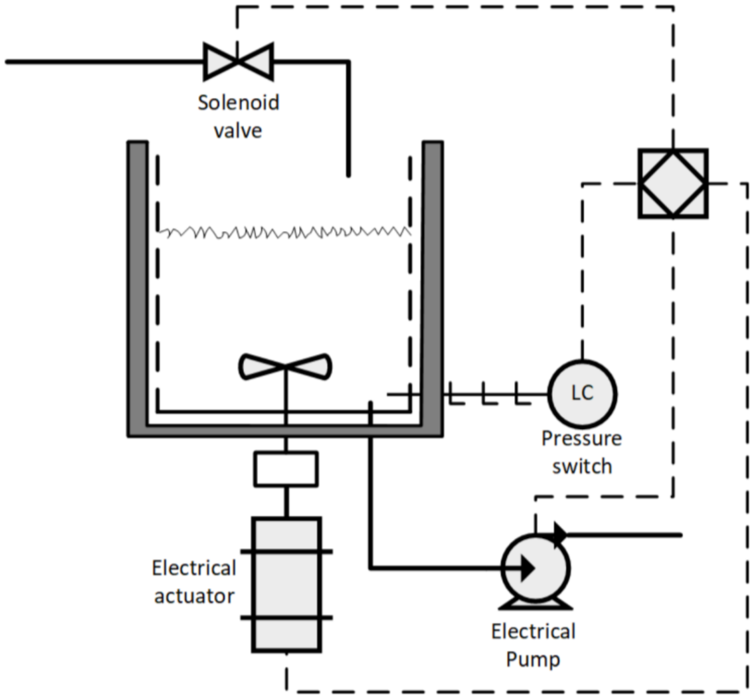


Figure 27 – Water hydraulic diagram of the washing machine

4.4.4.4. Refinement of information system

Mechatronic systems have as main characteristic the information processing including control, decision making, processing data, etc. Therefore, requirements related to these tasks can be considered and the Information System (IS) can have a first refinement. The results are both an architecture of the electronic and/or low power electric system and software requirements specification as necessary.

The architecture of the electric/electronic system can be a block diagram where the arcs represent electric variables. Main components such as power supply, emergency buttons, fuses, circuit breakers, drivers, relays, etc. are represented as well as auxiliary devices as output/input ports that are necessary. An example is shown in Figure 28.

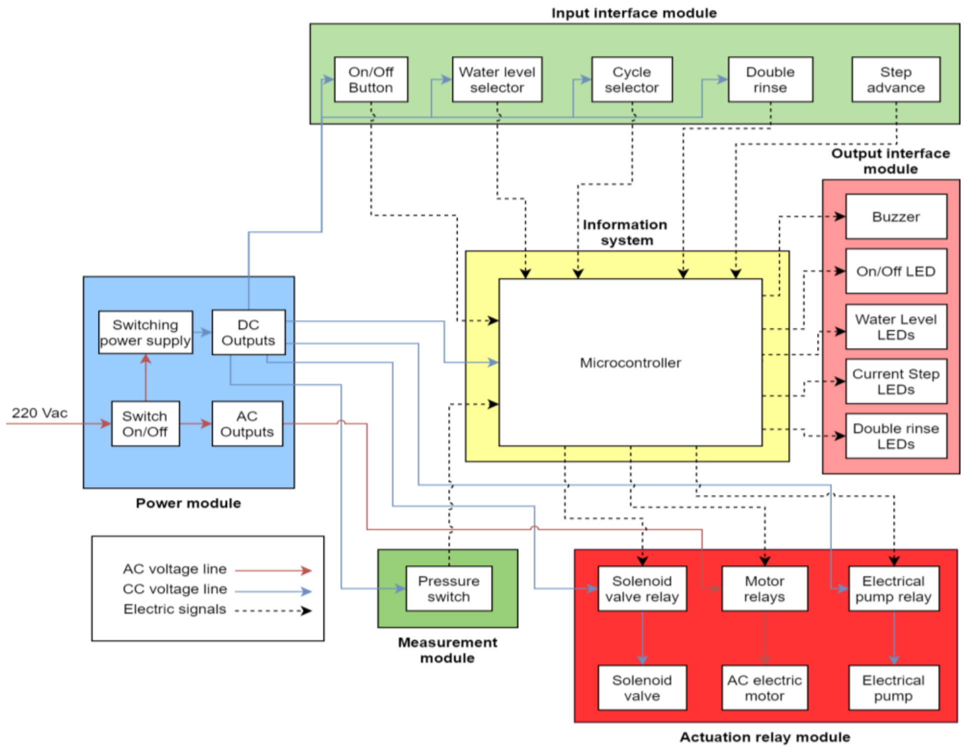


Figure 28 – Architecture of the electro/electronic system of the washing machine

Software development presents a particular methodology, typically involving the phases of analysis, design, and implementation (Jacobson *et al*., 1994; Jalote, 2005). The analysis comprises the following steps: Problem Analysis, Requirements Specification, and Requirements Validation. Correlating with the mechatronic design methodology presented hereafter, the Problem Analysis will be performed at the Planning and Task Clarification and the other two steps at the Conceptual design phase.

The result of the Analysis is the Software Requirements Specification (SRS) identified as one of the results of the task of refinement of information system as shown in Figure 16. It provides a formal documentation of what the software will do, comprising information about the software working structure, calculations that will be performed, data flow and manipulations, interactions with the user, and class diagrams (Jalote, 2005). Figure 29 presents the Entity-Relationship diagram of the washing machine, which is part of the SRS. The complete example of the SRS for the washing machine is presented in Section 5.13.

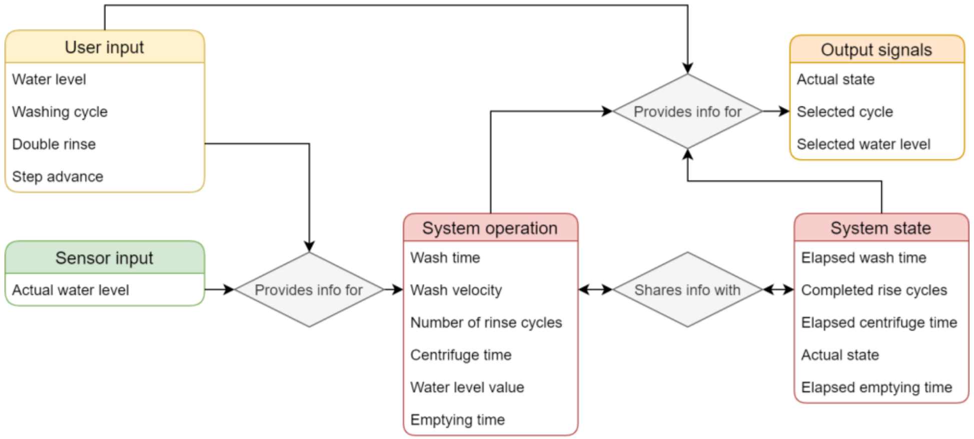


Figure 29 – Entity-Relationship diagram of the washing machine

4.4.4.5. Automation and Control Modeling

Electronic hardware and software implement the control of the mechatronic system. Therefore, the behavioral model of the system corresponds to the behavioral model implemented by the hardware/software.

Remembering that behavioral model describes when the functions are executed, at conceptual design level the system behavior model must describe when the functions performed by the operational means as well as components of the electric/electronic and software will be executed. Typically, it will be an event-driven model such as state machine, Petri net, flowchart or a Data Flow Diagram (DFD). Figure 30 shows an example of flowchart for the washing machine.

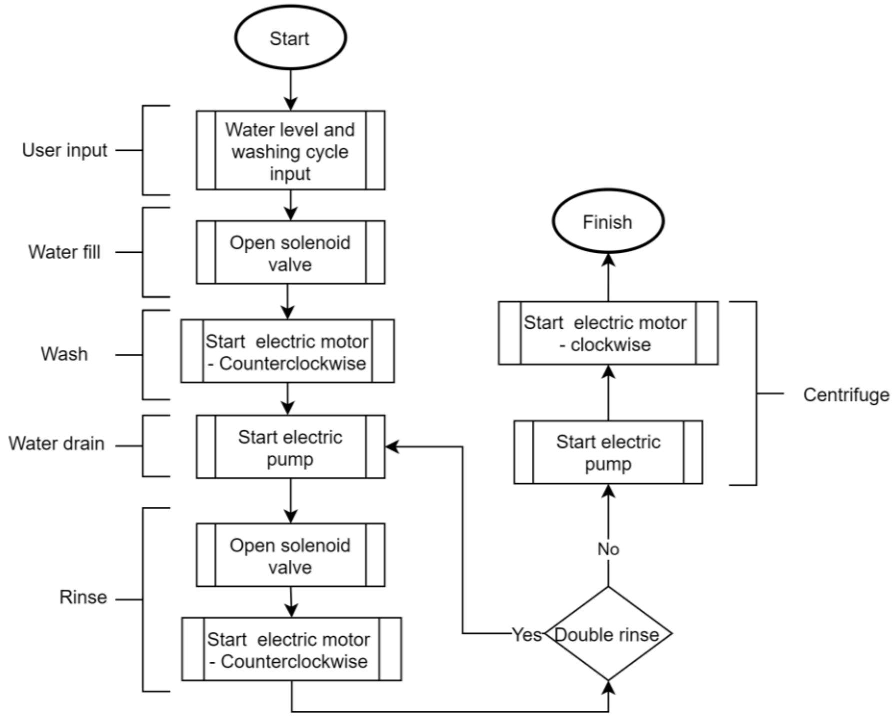


Figure 30 – Flowchart for the washing machine

The output of the synthesis of general structure and behavior is a set of models and diagrams that comprises the classical fields of a mechatronic system, the mechanical, electrical, and software domains. Therefore, the conceptual model is composed not just by a single drawing, instead, several diagrams and sketches are developed in order to provide a cross-domain solution concept, which will be the basis for the specific domain design and system integration of the following phases of the design process.

4.5. EMBODIMENT DESIGN

4.5.1. Introduction

Considering the design process model presented in Figure 12, the embodiment design phase is oriented for achieving a feasible solution for the system under design. Therefore, all parts of the system must be fully detailed in order to be possible to specify components for purchase, to manufacture and assembly pieces and components, to implement software and firmware, to simulate the component and system behaviors, and to build proofs-of-concept and prototypes.

The embodiment design starts from the conceptual model, resulting from the previous phase, that comprises the description of a technical solution, or a solution concept, represented by an initial understanding of the mechatronic system architecture where the main elements (operational means) are identified and interconnected. As it can be observed in Figure 16, the system architecture represented by Channel/Agency Net is the central model and the other models like a mechanical sketch, circuit diagram, electric/electronic block diagram, and software specification complement the functional and structural representation of the solution concept. Furthermore, the behavior of the mechatronic system, considering the main elements identified at that moment must be also described.

The embodiment design has the objective to refine the operational means identifying concrete subsystems necessary to accomplish the functions of measuring, actuation, input interfacing, and so on. At this moment, the design must be carried out at different technical domains according to the characteristics of the chosen subsystems. The physical and communication integration of the subsystems must be considered too in order to implement the material, energy, and information processing required in a mechatronic product.

In the presented design process model, the embodiment design phase (Figure 31) is divided in two steps:

- Definitive integrative design
- Domain-specific designs

These steps occur interactively while designs in each domain occur in parallel, as represented by the Petri net superimposed on the C/A net.

The Definitive integrative design is so-called because it must integrate all parts of the system in a consistent way. At the same time that the technical domains are identified from the definitive integrated design, this model is modified by the design decisions and detailed representations performed at each domain. This interdependence is represented in Figure 31 by both the arc from the mechatronic design architecture to the domain-specific design tasks and the arcs from the resulting definitive designs by domains to the integrative design step.

This approach implements an alternation between bottom-up and top-down design strategies as preconized in VDI 2206 (VDI, 2004) for more advanced design stages. Bottom-up design consists on combining separately developed subparts to form an overall system. Top-down design consists on firstly to gain knowledge of the basic structure and then to specify the elements of the structure more exactly by subsequent refinements.

Moreover, as shown in Figure 31, design data and specifications from the Planning and Task Clarification phase are used by the steps of the Embodiment Design. Technical and economic requirements are important inputs for component sizing and decision-making regarding the products to be used or the materials and manufacturing processes to be adopted. Information processing to be implemented must also follow client/user demands. Additionally, design requirements include behavioral demands such as actuation time responses, sequence of operations, tolerable response errors, etc. that are necessary only in the embodiment design phase. At this time, the effective performance of components, subsystems, and the full mechatronic product must be verified.

As an example, if the step Functional decomposition and means synthesis resulted on the use of an electric motor classified as an Actuation System, probably other components will need to be added such as a soft starter, protection devices, and so on. Therefore, the actuation system will comprise all these components in order to be able to receive an information (signal) from an Information System and convert electrical energy in mechanical energy. Based on this example, the aim of the embodiment design is to move from a working principle to an effective system or component, requiring detailing of the necessary complementary systems, their dimensioning, analysis of technical data sheets, etc. resulting on a complete description.

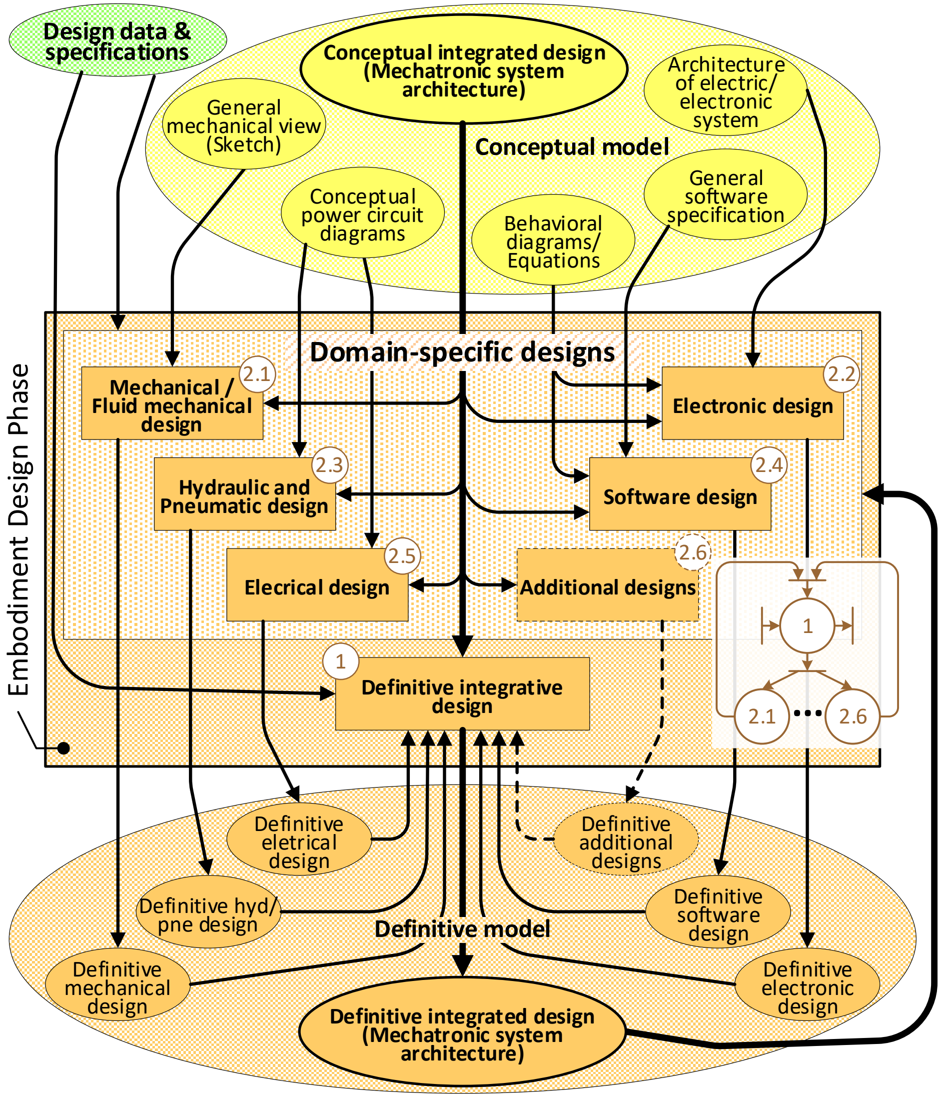


Figure 31 – Embodiment design phase

4.5.1. Integrative Design

4.5.1.1. Subtasks and their relationships

As stated in VDI 2206 (VDI, 2004), "a main feature of mechatronic products is the functional and spatial integration of components from the domains of mechanical engineering, electrical engineering and information technology." In the context of mechanical system design, Pahl and Beitz (2013) and Back *et al.* (2008) consider that through the embodiment design the overall layout design, including general arrangement and spatial compatibility, must be determined. It is also true for mechatronic systems in order to find the ideal physical disposition of components and subsystems. This issue impacts on product dimensions, ergonomics and shape, thermal and electromagnetic isolation between subsystems and with the external environment, maintainability, among other characteristics.

Taking into account the principles in the paragraph above, the **integrative design step** shown in Figure 32 is proposed. It uses results of previous phases which are design data and specifications and integrated conceptual design. Based on them and on representations in the fields of mechanics, hydraulics/pneumatics, electrical/electronics, software, among others initially obtained from the conceptual design, a **spatial integration** and **resource flow integration** can be started, resulting, respectively, on refined overall mechanical view and overall system architecture. Subsequently, comprehensive designs on each domain can also be initiated and their results fed back to the integrative design step.

Considering also the necessity for compatibility assessment related to resource flow and interconnection between components/subsystems, the four first subtasks in the integrative design are defined, as shown on the left of Figure 32. As represented by the Petri net on the top of this figure, there are two sequences of activities occurring simultaneously. The subsequent subtasks that will be discussed below occur sequentially ending with technical and economic analysis.

These first subtasks support structural and functional representations for the systems under design. Another very important issue for mechatronic system characterization is the **behavioral modeling**. Previously, in the conceptual design phase, a first general model of the system behavior was formulated, on that time mainly associating the sequence or logical correlation between the operating means identified on the mechatronic system architecture. Now, the parts of the system are very well established since deeply design tasks are performed in each domain, including construction/implementation of prototypes (virtual and/or physical). Therefore, behavioral models at each specific domain are formulated as will be discussed in Section 4.5.2.

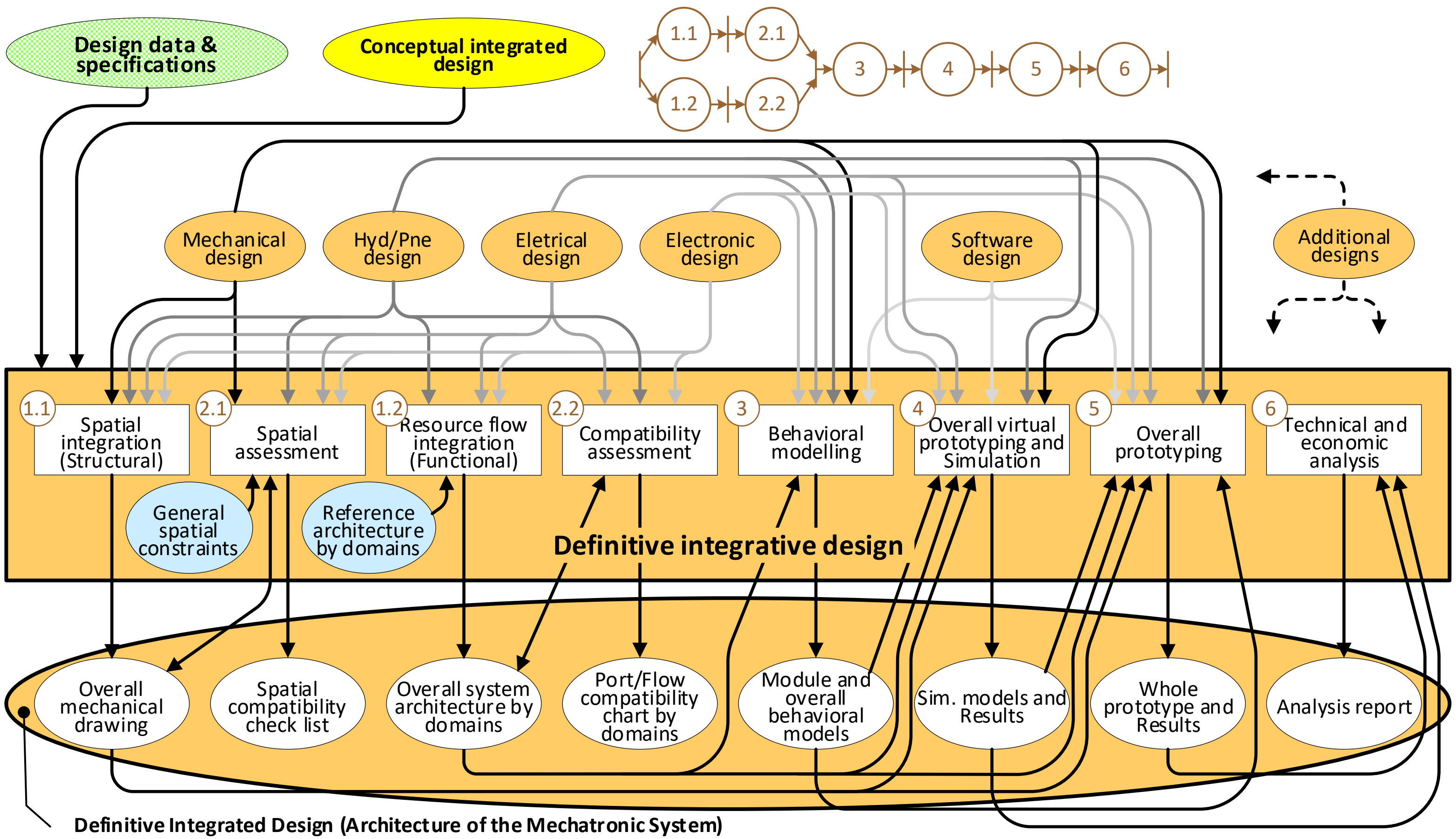


Figure 32 – Integrative design step

Depending on the magnitude of the product, it is usual to group some parts designed in different domains to constitute modules. Modules perform a macro function and, therefore, how this function is executed (that means, the module behavior) must be described too.

In this scenery, the objective of the behavioral modeling (shown in Figure 32) is to represent discrete-state and/or continuous-state descriptions (see Section 2.2.2) of the operation of these modules as well as of the overall system. For discrete-state representations, it is common to use event-driven models such as:

- State transition diagram (including Harel statechart)
- Place/Transition Petri Net
- GRAFCET
- Boolean equations and logic block diagram
- Flowchart, etc.

It is suggested to associate this type of model with a discrete controller that is part of the module or the overall system. Therefore, at the same time that the system operation is described it leads to controller implementation such that the controller outputs determine when actions take place. For example, discrete controller outputs could turn on or off an electrical motor, produce an alert output signal, fix a working piece, etc.

Otherwise, in simple systems, there is no implementation of an explicit controller. The interconnection of components can produce a logical dependence resulting on a sequential or combinatory operation of the system. For example, consider two electric motors driven by individual relays having multiple sets of contacts. The relay of the first triggered motor can switch on the second relay, that means, one can have a sequential operation without a control unit.

When continuous-state description is required, which is more usually associated with modules instead of the overall system, time-driven models are used, such as:

- Block diagram (with transfer functions)
- Differential equations
- Analytical expressions, etc.

Inversely to the discrete-state model, the continuous-state model is a behavioral representation of components and their interconnections and are used to predict the static and/or dynamic responses. The system does not necessarily include a controller (it can be operating in open-loop) but, if present, the controller structure affects the system response, however, it is not a representation of the system behavior.

Models of components, modules, subsystems, or the overall system can be implemented in software for computer simulation. It is very usual to perform one dimensional (1D) simulation using signal flow or power flow models resulting on the dynamic behavior of state variables. Manly for mechanical parts, 2D or 3D models are usual to give a spatial view of the system and the characterization of spatial variation of quantities.

3D models combined with human-product interaction model and perspective test related models can result on virtual prototypes. According to (Wang, 2002), "Virtual prototype, or digital mock-up, is a computer simulation of a physical product that can be presented, analyzed, and tested from concerned product life-cycle aspects such as design/engineering, manufacturing, service, and recycling as if on a real physical model. The construction and testing of a virtual prototype is called virtual prototyping (VP)."

As included in the integrative design (Figure 32), the subtask **Overall virtual prototyping and simulation** includes the construction and running of one or more virtual prototypes and execute of 1D and 2D simulations of the whole system. As represented in that figure, models built in the different domains and the overall system architecture representation are grouped in order to accomplish this task.

After the system performance has been evaluated by simulations and/or virtual prototypes and eventually some parts or components redesigned to achieve the requirements, a **physical prototype** can be constructed. In mechatronic systems, it comprises the integration of parts or modules from the several technical domains, Software (and firmware) implementation is also taken into consideration.

Through the prototype, test and evaluation of form, performance, design fit, user-interface, and manufacturability are carried out. Design concepts can be compared and proved and also the product prototype can be presented to the client or future users. Prototyping parts of the system is also useful for validating mathematical models so that the models can effectively support decision-makings.

The final subtask corresponds to the **technical and economic analysis**. As stated by Pahl and Beitz (2013), the embodiment design is characterized by repeated deliberation and verification related to aspects like safety, ergonomics, production, assembly, operation, maintenance, recycling, costs, and schedules. Technical and economic requirements were established in the begging of the design process and it is mandatory to verify if they were accomplished. It can be done by analyzing the results from simulations and virtual and physical prototyping.

4.5.1.2. Spatial constraints

It consists of a checklist to make sure that the physical constraints are being met. The most common physical constraints are related to the system dimensions and geometries, such as maximum width, height, and length, the existence of sharp edges or nonergonomic geometries on handling surfaces, and unbalanced weight on rotating mechanisms. Some spatial constraints might result from the technologies and products that will be used in the mechatronic system, for instance, when a variable-frequency drive is applied to control an AC motor speed, it might introduce high amounts of noise in the measurement system, which requires specific positioning and shielding of the drive. Motor drivers usually require a specific positioning and distance from other drivers in order to allow the heat dissipation, for instance, the fin orientation frequently is vertical to allow the natural air convection through the drive. Another example is when strong magnetic fields are involved on the system, such as the magnetic resonance imaging devices. In this case, every component with magnetic or ferromagnetic materials needs to be carefully allocated and fixed within the machine.

The spatial assessment can be carried out by analyzing the design requirements related to physical aspects of the system, including dimensions, geometries, ergonomics, packability, transportability, and so on. Also, specific guidelines for physical installation and spatial constrains can be found on the component's datasheet.

4.5.2. Domain-Specific Designs

4.5.2.1. Introduction

As discussed before, the first configuration of the architecture of mechatronic system, named conceptual integrated design, was created in the conceptual design phase. It must be improved in the embodiment design moving from abstract arrangement of operational means (measuring systems, actuation systems, etc.) to concrete arrangement of modules and/or components. Moreover, these subsystems must be refined in order to achieve executable designs that will be implemented by hardware or software. As typical of mechatronic systems, modules/components with distinguished technological domains will be used and, consequently, separated work teams and/or approaches are necessary.

Therefore, once the first version of the definitive integrated design is reached, the domain-specific design step takes place. As each project evolves, the definitive integrated project must be revised and its changes fed back to the specific domains again.

As it can be observed in the domain-specific designs presented in the next sections, a conceptual design activity is expected to occur since a new component, assembly, or piece of software will be developed. It means that steps, tasks and subtasks discussed in Section 4.4 must be carried out resulting on a complementary conceptual model for each specific domain.

The next sections detail the main subtasks for the most common technology domains found in mechatronic systems: mechanics, hydraulics and pneumatics (fluid power systems), electric, electronics (hardware and firmware), and software. Depending on the nature of the mechatronic system, additional specific designs may be required, such as vision technology, optoelectronics, industrial processes, etc.

Importantly, design process models do not represent a sequential or logical/conditional arrangement of activities. Being a Channel/Agency net, they show the activities (subtasks), the received resources needed to carry them out and the resulting resources. Double arrows mean that a resource is used by an activity but is also updated. However, there is an intrinsic causality, since it is necessary to have a certain resource to carry out the activity. Therefore, the arrangement of the subtasks in the diagram indicates that there is a preferred sequence, but there is no restriction on reworking the subtasks. It's an evolutionary design process.

4.5.2.2. Mechanical domain

Figure 33 presents a general process for designing mechanical elements, components, mechanisms, mechanical transmissions, and so on. Analysis under the point of view of solid mechanics and fluid mechanics can be carried out as necessary.

When designing components with functional principles from other domains, the design must be carried out under the point of view of mechanics and the other area like fluid power, optoelectronic, electric, and so on. It includes the design of pumps, motors, generators, valves, switches, and all components that must be mechanically shaped and sized. The typical subtasks of mechanical design are:

- **Mechanical drawing:** The aim of this subtask is the development of 2D and 3D models of the mechanical parts. The general arrangement and spatial compatibility shall be evaluated, as well as the shapes and materials of the components. The model will iteratively be updated according to the design constraints that usually appear in the following steps of the mechanical design domain. The 3D models will result in scale drawings, which are detailed design with the dimensions of the components.

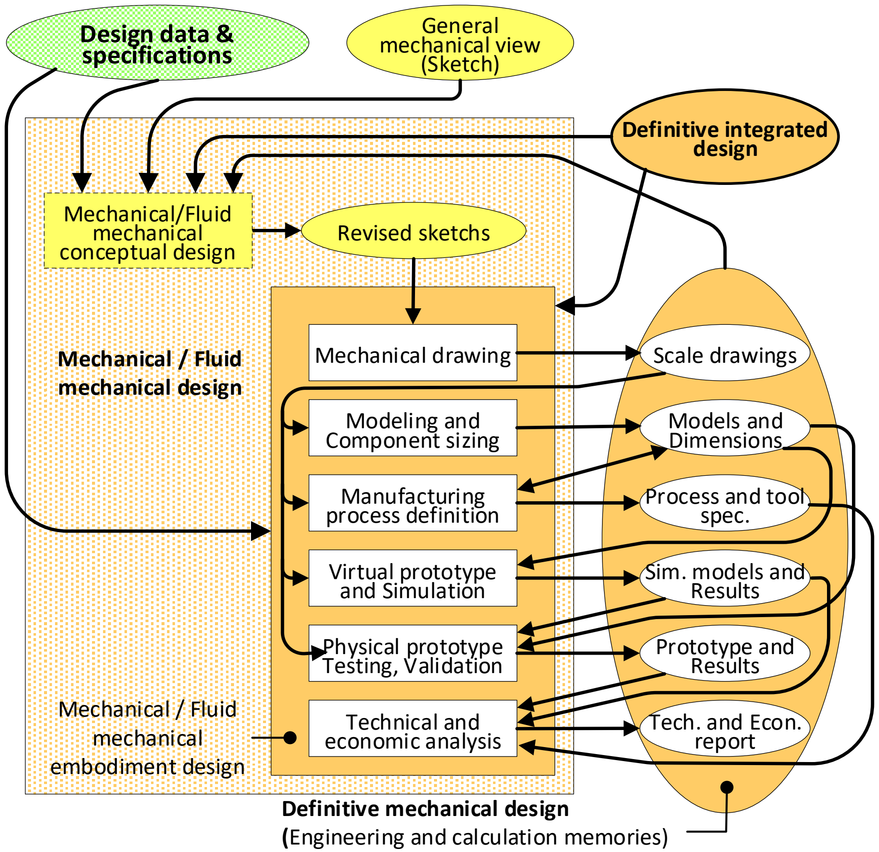


Figure 33 – Design in the mechanical domain

-**Modeling and component sizing:** At this point the mechanical parts such as shafts, beams, bearings, and gears will be sized and modeled according to the design requirements. The sizing shall be carried out based on the classical mechanical engineering design literature.

- **Manufacturing process definition:** Based on the component dimensions, geometry, function and available budget, a suitable manufacturing process must be defined. Simple components which do not perform critical tasks might be manufactured with conventional processes and tools (saw cutting for instance). However, more critical and complex components require more expensive processes such as machining, welding or casting.

-**Virtual prototype and simulation:** It is an alternative to evaluate the performance of the components before manufacturing it. The classical tools for mechanical simulation include computational fluid dynamics (CFD), finite element analysis (FEA) and dynamic simulation (differential equations).

- **Physical prototype, testing, and validation:** It is a preliminary and rough physical version of the mechanical components. The main advantage of prototyping is the possibility to present a realistic three-dimensional visualization of the final product for the clients and investors, as well as reduce the risk of design errors by the detailed physical analysis of the prototype. The most common techniques of mechanical prototyping are 3D printing and 2D laser cut materials. However, other manufacturing processes can be used in order to construct a prototype of the complete systems or parts of it.

- **Technical and economic analysis:** The solutions developed in the mechanical design must be evaluated for technical and economic feasibility. Finally, the knowledge and lessons learned during this design task must be properly documented.

4.5.2.3. Hydraulic and/or pneumatic domain

The design of hydraulic and/or pneumatic systems follows a set of subtasks as shown in Figure 34. Since that a fluid power circuit usually operates integrated with continuous or discrete controllers, the modeling and analysis should include these subsystems too. In most cases, controllers are implemented by electric and/or electronic devices such that their design is carried out at the corresponding domain.

This task includes not only the design of circuits but also components that include the mechanical structure and some times electrical/electronic devices such as embedded sensors, electromagnetic actuators (solenoids, torque-motors), etc. In these cases, the design occurs in the different domains simultaneously.

The main subtasks are presented below:

- **Circuit diagram design:** The fluid power diagram consists of a symbolic representation of actuators, valves, accumulators, pumps, reservoirs, the connection of such components through pipes and hoses, and also the sensors related to these components. The fluid power diagram shall be carried out according to the ISO 1219 standard series (ISO, 2012a), (ISO, 2012b), (ISO, 2016).

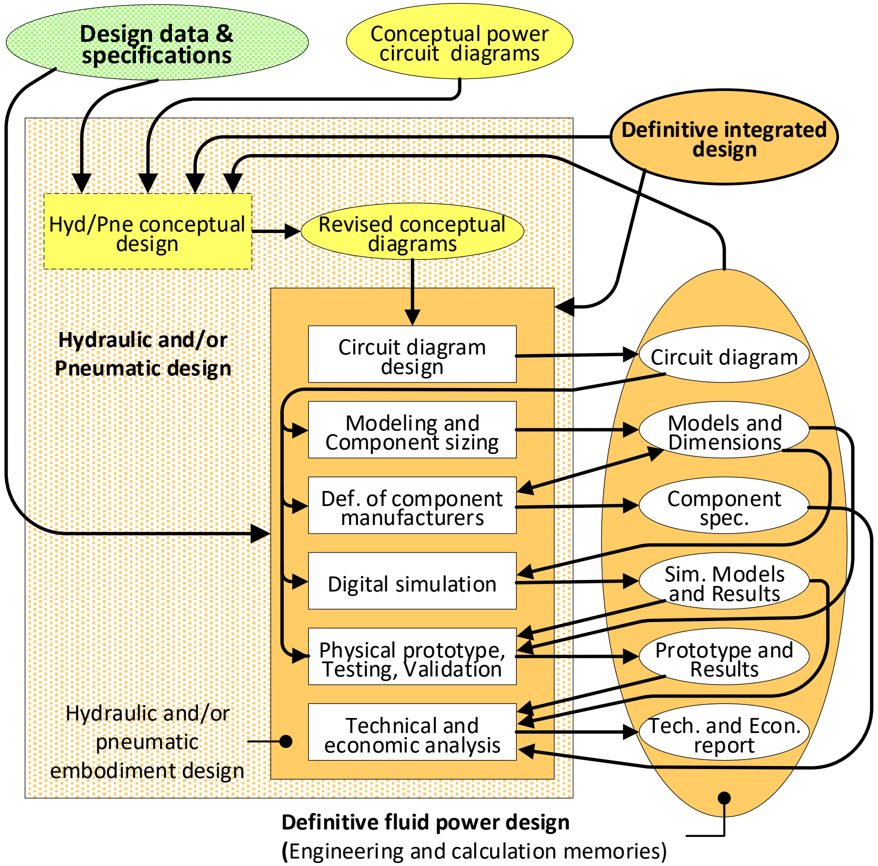


Figure 34 – Design in the fluid power domain

- **Modeling and component sizing:** The components of the fluid power system shall be sized according to the design requirements and it might be carried out based on classical literature. After preliminary sizing (mainly in steady-state), static and dynamic simulations are an important tool to be considered during this process, since it may result in an improved design, with higher system efficiency and performance.

- **Definition of component manufacturers**: After the first sizing procedure, a suitable commercial version of the designed components has to be chosen. It is important to look for local suppliers and/or partnerships that the company might have. Usually, the selected component will have different characteristics than the designed one, thus the sizing process has to be reevaluated in order to check if the design requirements are still being met.

- **Digital simulation:** It is an important step in order to understand the whole fluid power system behavior, mainly in the case of more complex arrangements. The simulation helps to evaluate the system performance and to identify design errors. It is usually an iterative process between dimensioning the component, defining the component manufacturers, and simulating the system. The fluid power simulation might be static, which does not consider transient behavior such as pressure and velocity varying in time, or dynamic, where the transient behaviors are represented according to the differential equations that govern the system. There are several simulation environments that are capable to perform static and/or dynamic simulations.

- **Physical prototype, testing, and validation:** Fluid power systems might be physically tested in a controlled environment before being installed in the real system. The advantage of prototyping fluid power systems is the possibility of experimentally evaluating and improving the performance and behavior of the system, since many features are not feasible to be simulated. To carry out a fluid power prototype, it is necessary to design a suitable mechanical structure, select the necessary sensors and define a control system software and hardware to perform the data acquisition and control. A prototype is also important to validate simulation models of parts or the whole system in order to support the behavior understanding.

- **Technical and economic analysis:** The solutions developed for the fluid system must be evaluated according to the technical and economic requirements established in the first phase of the design. In case of failure to meet some requirements, the previous subtasks must be reworked.

4.5.2.4. Electrical domain

The design process of electrical systems is similar to fluid power systems, however, based in different physical laws and constructive principles. In the case of electrical circuits, digital simulation and physical prototypes are only needed in the development of non-trivial solutions, since the behavior of typical circuit components is very predictable.

The subtasks presented in Figure 35 and described below also apply to the design of electrical components; however, the emphasis is more on the selection and choice of ready-to-use components, as in electrical control and power circuits, than on the definition of shapes, materials, elementary components, and so on. The subtasks are:

- **Circuit diagram drawing:** The wiring diagram consists of symbolic representations of the electric circuit components, including motors, relays, breakers, contactors, wires, and other electrical components. The focus of this diagram is the command and power system and therefore the design of electronic devices is not included except their interconnection with other electrical components.

- **Modeling and component sizing:** The components of the electrical system shall be sized according to the design requirements. The sizing might be carried out based on classical literature and in observance of the manufacturer guidelines.

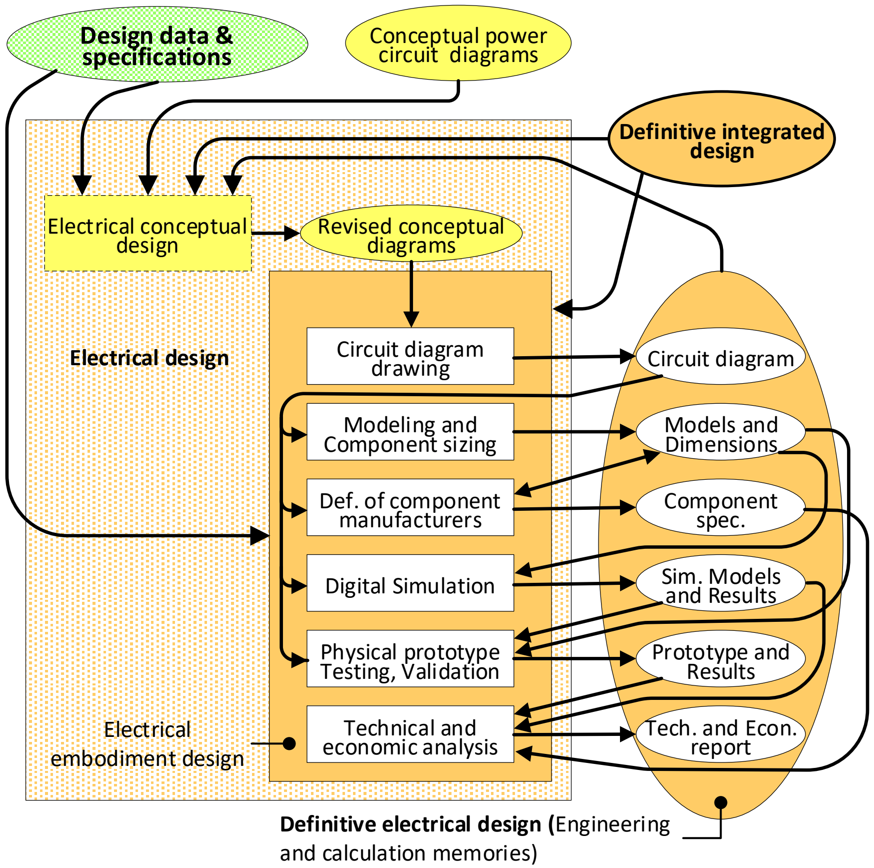


Figure 35 – Design in the electrical domain

- **Definition of component manufacturer:** Commercially available components shall be selected after the first sizing procedure. It is important to look for local sellers, company partnerships and also internet sales. It is common that the selected component does not have the same characteristics as the designed one. Therefore, the sizing process has to be reevaluated to check if the design requirements are still being met.

- Digital simulation: System simulation helps to understand system behavior and performance, as well as to identify design errors. It is generally an iterative process between component sizing, component manufacturer definition, and system simulation. Electric system simulation might be static, where the current and voltage variations are not considered, or dynamic, where the transient responses are analyzed according to the differential equations that govern the system.

- Physical prototype, testing, and validation: Since some characteristics of the electrical system may not be feasible to model, system prototyping is an important subtask to experimentally evaluate the system and verify that it performs according to the design requirements. In order to build an electrical system prototype, it is necessary to design a suitable mechanical structure for assembling the system components and, if necessary, select the sensors and control system to perform and record the tests.

- Technical and economic analysis: The solutions developed for the electrical system must be evaluated according to the technical and economic feasibility. At the end, the knowledge and lessons learned during this design task must be properly documented.

4.5.2.5. Electronic (hardware and firmware) domain

The design process for embedded software and hardware systems presented herein is based on the double roof model proposed by Jürgen Teich (Teich, 2001) for embedded hardware/software systems. However, instead of representing in the original format, it is drawn using Channel/Agency net, but the elements and the design process are the same as proposed by the author.

It consists of a concurrent design of hardware and software as one can see in Figure 36. The right side corresponds to the software design process, while the left side to the hardware design process. In the middle, the common activities applied for both hardware and software are represented. The Prototyping and Technical and economic analysis subtasks are not part of the original model, however, they were included here to follow the mechatronic system design principles used in the present work.

The process consists of a sequence of synthesis steps, each of which is a translation of a behavioral description at a given level of abstraction into a structural description of the same level. Through a synthesis step, a specification is transformed into an implementation.

In Figure 36, the ellipses at the top correspond to specification models associated with:

- **Electronic system level (ESL):** Behavioral system specification (e.g., algorithms, tasks).
- **Software development:**
  - Process networks, tasks: Include process networks, task-level graphs, languages with support for threads (e.g., Java, etc).
  - Procedures, functions: Programs, functions, procedures as encountered in high-level languages.
- **Hardware development:**
  - Arithmetical and logical functions: Coarse granular arithmetic and logical functions.
  - Boolean equations, FSM, HDL: System of Boolean equations or finite-state machine (FSM).

At the bottom of this figure, the definitive electronic design comprises the resulting implementation models, which are:

- **Software development:**
  - Instruction set architecture: uni-processor or a multi-processor architecture in software.
  - Microarchitecture: instruction-level of the target-processor on which the code is to be executed.
- **Hardware development:**
  - RTL blocks: communicating functional blocks.
  - Netlist: textual representation including logic gates and registers.

The process comprises five levels of abstraction that are (Gerstlauer *et al.*, 2009; Teich, 2012): System, Architecture, Logic, Module (task), and Block (instruction) and they occur by the synthesis steps (crosshatched rectangle) shown in Figure 36.

Information about the implementation at a certain level is transferred to the next lower level of abstraction as additional specifications or constraints. It is represented by the subtask with squared hatching.

Therefore, the subtasks of the electronic embodiment design are those associated with the synthesis steps, information transfers, prototyping, and technical and economic analysis. They are described below grouped into hardware and software, hardware, and software designs.

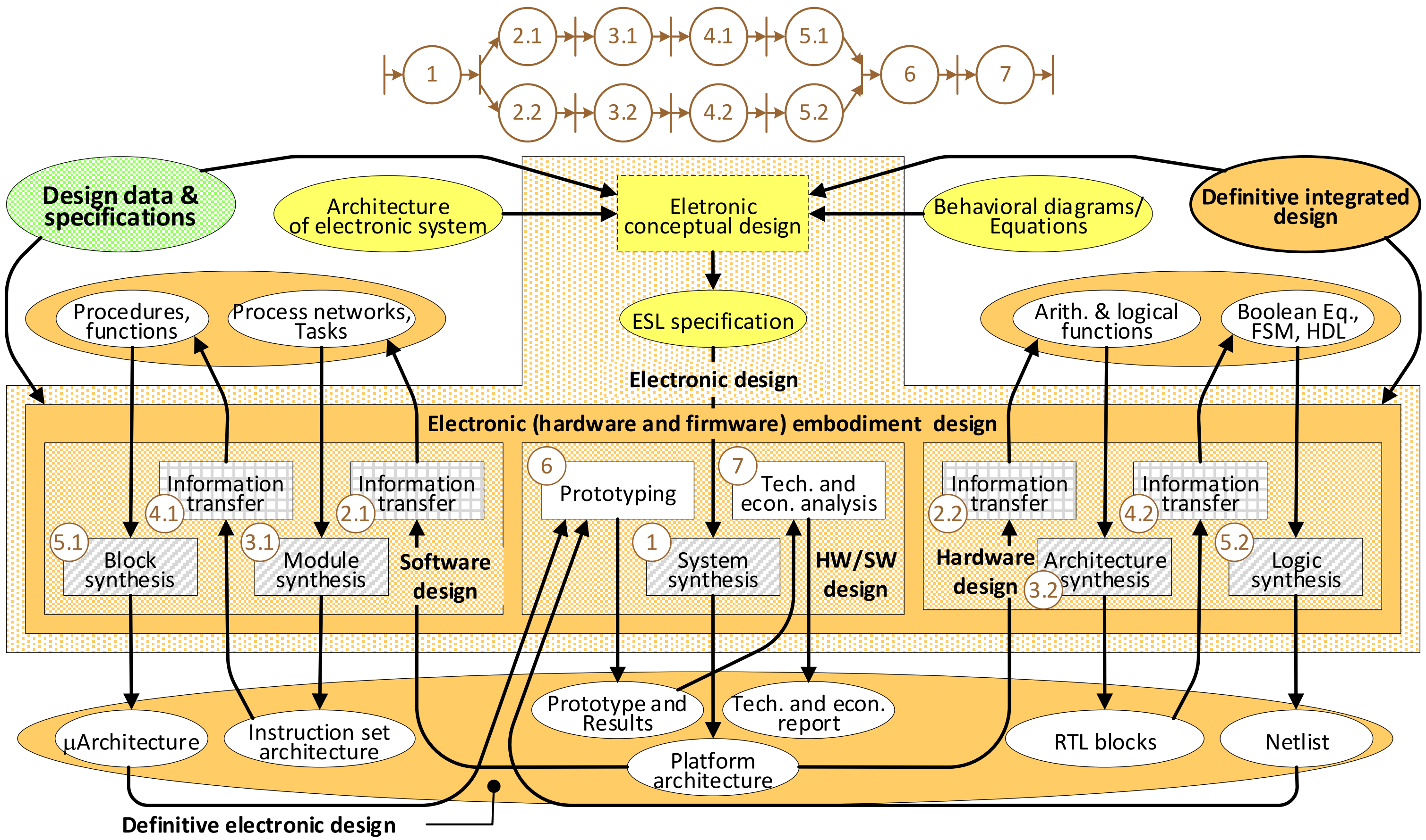


Figure 36 – Design in the electronic domain

Applied for hardware and software:

- System synthesis (Hardware/software partitioning): The aim of this activity is to transform the electronic system level (ESL) specification represented by a behavioral model that is often some kind of network of processes communicating via channels. It corresponds to a functional and/or behavioral specification of the whole system (either model-based or language-based). Additional input to this activity is also the design specifications including mapping constraints and implementation constraints (maximum area, minimal throughput, etc.). The result is a network of communicating subsystems, i .e., a platform model at ESL that is typically a structural model consisting of architectural components such as processors, subalgorithms, busses, dedicated hardware units, memories, etc. In this step, one cannot yet distinguish between hardware and software.

- Prototyping and testing: Due to the low cost of most of the electronic components, the physical prototype is a usual process during the electronic device development. With the prototype, it is possible to analyze a real working version of the electronic system. In order to build the electronic system prototype, it is necessary to select a suitable breadboard, jumper wires and DIPs (dual inline package) integrated circuits.

- Technical and economic analysis: The solutions developed for the electronic system shall be evaluated according to the technical and economic feasibility.

- Information transfer: Subtask passing information about the implementation at a certain level of abstraction directly to the next lower level as an additional specification information or constraints. These activities occur between synthesis activities in both hardware and software sides.

Applied for hardware design:

- Architecture synthesis: arithmetic and logical functions representing processes selected to be implemented as hardware accelerators are synthesized down to an RTL (register transfer level) description in the form of controller state machines that drive a datapath consisting of functional units, register files, memories, and proper interconnect. RTL blocks are communicating functional blocks that implement coarse granular arithmetic and logical functions.

- Logic synthesis: This activity transforms a given specification of a system of Boolean equations or a finite-state machine (FSM) provided in the form of either a table, diagram, or alternatively hardware description language (HDL) specification in a netlist implementing this FSM by choosing variable encodings, applying logic minimization, and finally allocating logic gates and memory elements from a library. Netlist is a description of the connectivity of an electronic circuit. In its simplest form, a netlist consists of a list of the electronic components in a circuit and a list of the nodes they are connected to.

Applied for embedded software (firmware) design:

- Module (task) synthesis: Communicating processes/threads bound to the same processor is translated into the instruction-set architecture (ISA) of the processor, targeted toward and running on top of an off-the-shelf real-time operating system (RTOS) or a custom-generated runtime environment. This step involves the generation of source code in a target programming language for subsequent code compilation.

An instruction set architecture (ISA) is an abstract model of a computer. It is also referred to as architecture or computer architecture. A realization of an ISA, such as a central processing unit (CPU), is called an implementation.

- Block (instruction) synthesis: Comprises the refinement of behavioral models including programs, functions, procedures as encountered in high-level languages to the instruction-level of the target-processor on which the code is to be executed. The instruction set of programmable processors is realized in hardware by implementing the underlying microarchitecture.

4.5.2.6. Software development:

Based on the literature of software engineering and object-oriented approach (Dathan & Ramnath, 2015) (Jalote, 2005) (Jacobson *et al.*, 1994), the tasks related to software development are described next. Software development embraces not just the design itself, but also implementation, testing, integration, installation, and maintenance. The development is much more interactive than mechanical, fluid power, and electrical designs, resulting on the execution of tasks repeatedly, including code implementation from the beginning of the development.

As presented below, the tasks associated to the software area are correlated with the design phases for mechatronic systems used in the present work.

- Requirements analysis (fits in task clarification design phase): Extracting the requirements of a desired software product is the first task in creating it. Requirements are gathered and a use cases document, functional requirements document, and a non-functional document (functional requirements written in natural language) are obtained. The use cases are formulated conjointly with the stakeholders (i.e., client, specialized people, etc.), through brainstorming or by reading documentation about the domain. These documents comprise the *software requirements*.

- Software (system) analysis (fits in the conceptual design phase): Aims to describe formal and completely the system requirements using the developer language. Includes the use of use case diagrams, class diagrams, sequence diagrams, etc., enabling the identification of some of the components of the system and the relationships between them. The end product of this activity is the *software specification*, corresponding to a conceptual model for the system which describes the functionality of the system, identifies its conceptual entities and records the nature of the associations between these entities.

- Design (fits in the embodiment design phase): In this activity, the structure of the system is defined. In object-oriented programing, classes are defined, their attributes, their methods, and the relations among them. This step is followed by determining the software and hardware structures needed to implement the functionality discovered in the analysis (system specification) stage. The selection of a programing language takes place, which shall be suitable to perform the design requirement tasks, have an available programming environment, and be familiar with the programming team. Pseudo code is developed, which is a detailed description of what an algorithm must do. The pseudo code is expressed in a natural language such as text description and/or block diagrams.

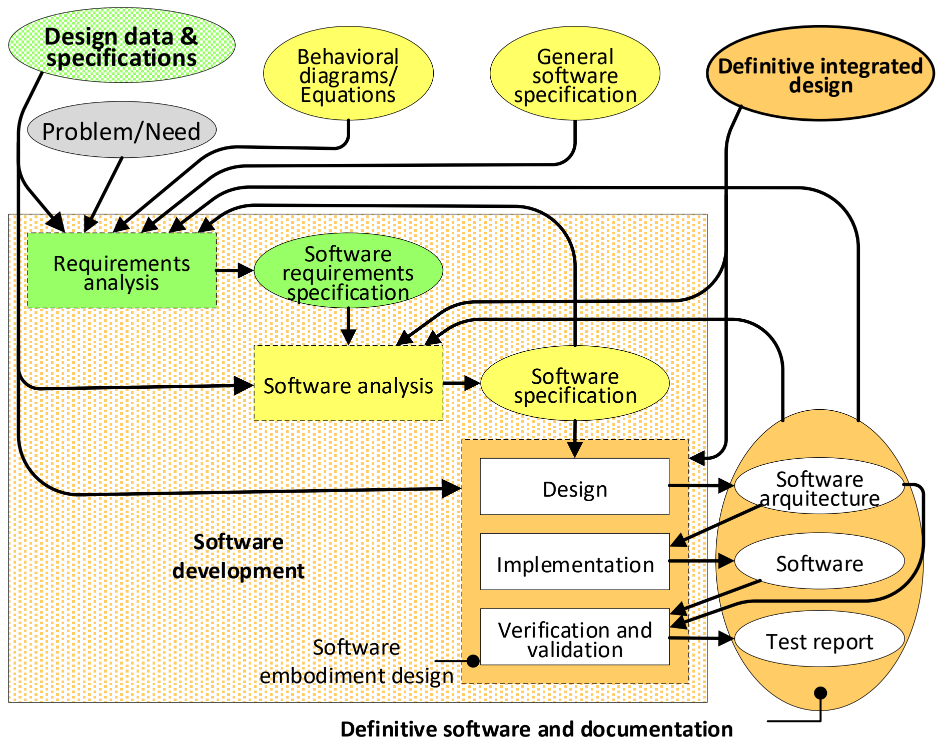


Figure 37 – Design in the software domain

- Implementation (fits in the embodiment design phase): consists of code writing and debugging. Lower-level issues and language features employed are discussed.

- Verification and validation (Testing) (fits in the embodiment design phase): after the code is written and compiled, it usually is tested along with the electronic system prototype, the sensors and, if possible, the actuators as well. The implementation is checked against the specifications This phase can occur iteratively with the implementation phase, with the purpose of developing correctly the application while checking if it is in concordance with its specifications.

4.6. DETAILED DESIGN

When the embodiment design is completed, the phase of optimizing the specifications of the components begins. This process is based on the results obtained from the first prototype developed during the embodiment design phase. When this initial prototype of the complete system is tested, it is common to observe unforeseen behaviors that may deviate from the desired system performance. According to Back *et al*. (2008), several analyses are carried out during prototype testing, including the evaluation of the system and component safety. As a result of the prototype evaluation, optimizations to existing specifications and even the definition of new specifications may be required, leading to updates in both the system requirements and the domain-specific designs. This iterative process continues until the prototype is fully validated and approved.

According to Pahl *et al*. (2007), one of the most important aspects of the detailed design phase is the preparation of production documents, including component drawings, assembly drawings, and the appropriate part list. Considering the multidisciplinary nature of mechatronic systems, the design within each specific domain implies in the detailed specification of its subsystems, aiming the selection of suppliers, determination of manufacturing and assembly processes, definition of quality control procedures, and the obtaining of required certifications.

In the design process model presented in this book, the detailed design phase (Figure 38) is structured into two steps, which take place concurrently and interactively:

- Specification detailing
- Global design

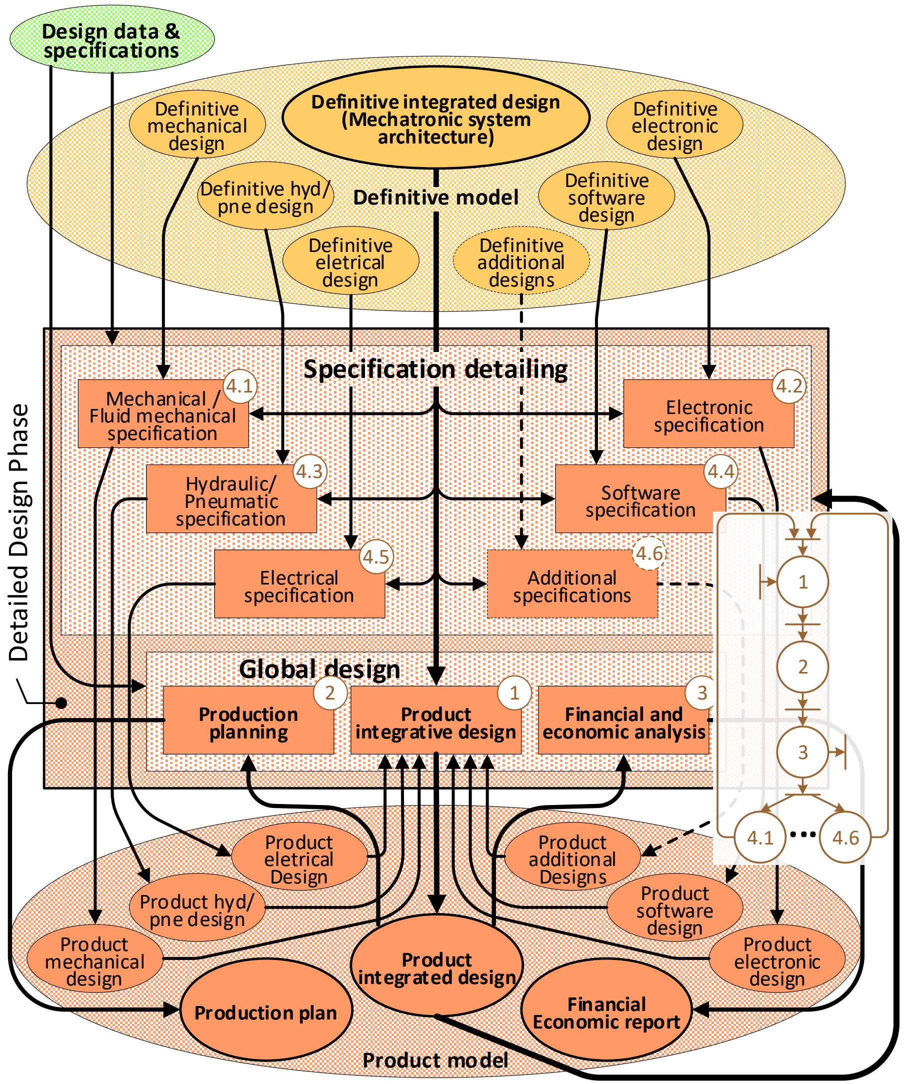


Figure 38 – Detailed design phase

The main aspects that characterize the **specification detailing step** are highlighted below:

- **Mechanical/Fluid Mechanical Specification**: Manufacturing methods (machining, casting, forging, injection molding, 3D printing, etc.); dimensional and geometric tolerances (e.g., ISO 286 system); assembly and joining considerations (welding, bolting, press-fitting); heat treatment and hardening (annealing, tempering, carburizing, nitriding, etc.); surface treatments (anodizing, painting, powder coating, etc.); quality control; cost; and production volume.

- **Hydraulic/Pneumatic Specification**: Component suppliers and availability (to select reliable suppliers with stable lead times and post-sales support); assembly aspects (fast and intuitive assembly, easy access to serviceable components, labeling of ports and hoses); maintenance aspects (availability of maintenance kits, integration of sensors for predictive maintenance); sealing materials; component coatings (chrome plating, anodizing, corrosion-resistant painting); and lifecycle considerations for actuators, valves, pumps, and hoses.

- **Electrical Specification**: Component standardization (off-the-shelf components from reliable suppliers, preference to industrial standards for connectors, fuses, and relays); safety and protection (specification of grounding and shielding, components for overvoltage, overcurrent, and reverse polarity protection); wiring characteristics (cable voltage and current ratings, color coding, labeling); testability (inclusion of test points or diagnostic connectors); and electromagnetic compliance (minimizing noise emissions, use of shielded cables).

- **Electronic Specification**: Component supply chain (selection of reliable suppliers, preference for industry-standard components); PCB design for easier manufacturing and assembly (PCB shape that fits the mechanical envelope, mounting holes, robust connector points); maintenance aspects (LEDs for diagnostics, support for remote diagnostics).

- **Software Specification:** Communication protocol (Modbus, Profibus, Profinet, CANopen, etc.); input/output and sensor integration (specify sensor types, update rates, filtering algorithms); fault tolerance (define watchdog timers, fault detection, fail-safe behaviors); update strategy (define the firmware update mechanism); security and access protocols (define access levels, firmware authentication, and encryption); maintainability and scalability (use consistent coding standards and documentation, plan for future updates, adopt a version control platform).

The **global design step** involves three tasks, the first of which is the **Product Integrative Design** (Figure 39). Most of its subtasks are similar to those in the definitive integrative design (Figure 32), mainly consisting of updating the results from the embodiment design. However, during the detailed design phase, the focus is on achieving a final product demonstration that addresses all the unforeseen behaviors observed during the first prototype testing. Additionally, it includes the development of all product documentation, such as testing and validation reports, product manuals, lifecycle reports, and regulatory and compliance documentation.

The second task of the Global Design phase is the **Financial and Economic Analysis**. This analysis should encompass detailed cost estimations, including component sourcing, manufacturing processes, assembly labor, testing, quality control, and logistics. Break-even analysis, return on investment (ROI), and total cost of ownership (TCO) provide valuable insights into the economic sustainability of the product, as well as support for future decision-making processes.

The third step of the Global Design phase is **Production Planning**. This step involves translating the detailed outputs of each domain-specific design into manufacturing workflows, including the development of assembly sequences, definition of workstations, selection of manufacturing tools, and allocation of human and technical resources. This step also requires establishing end-of-line testing to ensure that each unit meets user and design requirements, regulatory standards, and the definition of quality control analyses.

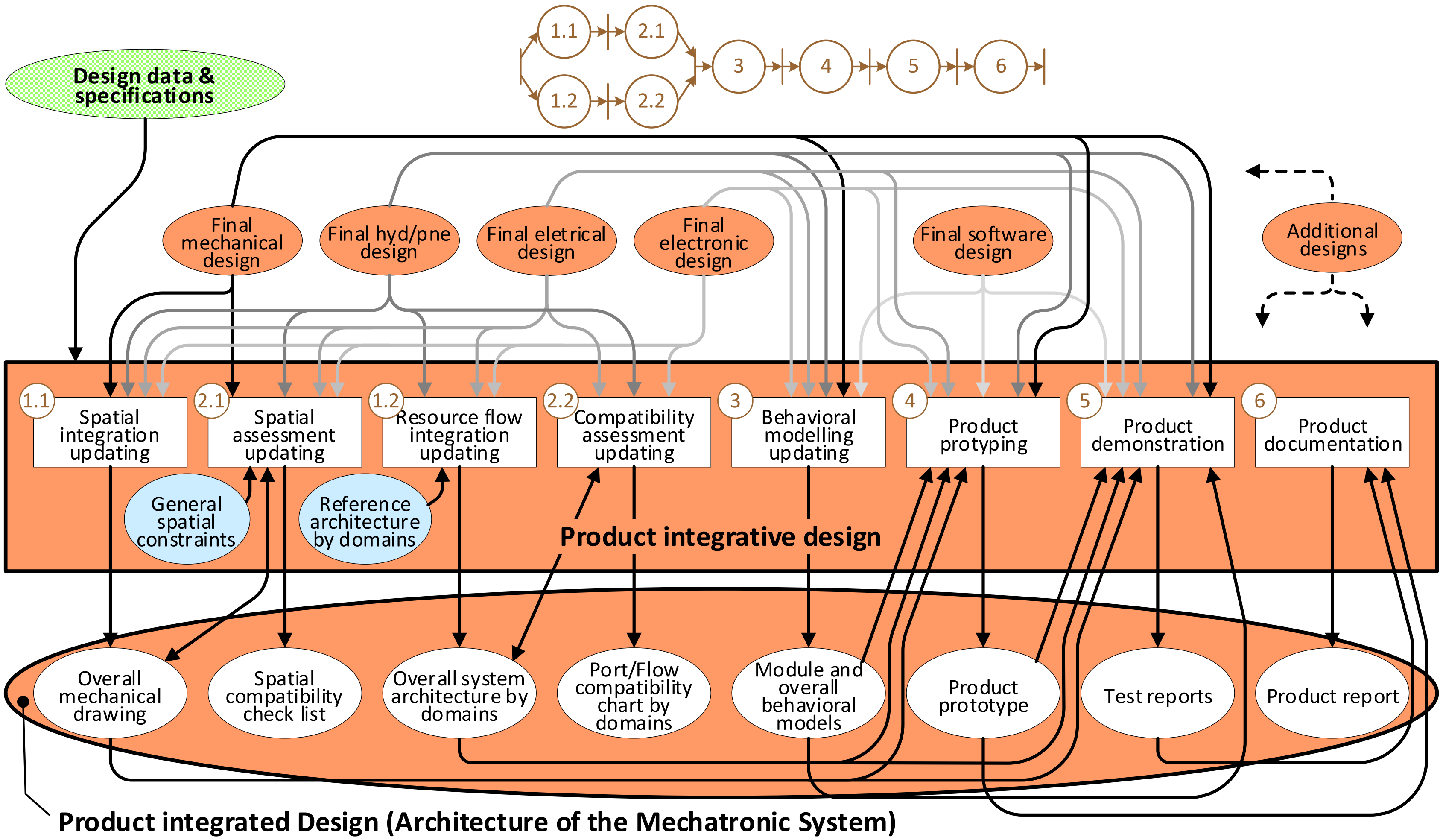


Figure 39 – Product Integrative Design

5. USE CASE: AUTOMATIC WASHING MACHINE

In this section, a hypothetical scenario will be created in order to present an example of the design process of a washing machine. In this scenario, a company called XYZ identifies a market opportunity in the household appliances sector. The opportunity is due to the high prices of the available washing machines, which are imported from other countries. Even though some figures and diagrams of the washing machine design process were already resented in section 4, they will be presented again in this example in order to follow a logical sequence.

5.1. PRODUCT PRELIMINARY IDEA

The XYZ company needs to develop a washing machine in order to become a new competitor in this market. Since there are many imported models available, the project aims to develop a national technology, considering the technology available worldwide.

5.2. SUBJECT BACKGROUND

5.2.1. General process

The need to wash clothes began when the people started to wear fabrics instead of animal skin. Initially, only clean water was used along with manual movements of agitation and twisting, which removed the dirt from the fabric. However, over time people began to notice that the addition of a few substances helped to remove the dirty, originating the soap as it is known today (Svitavy Museum, 2020).

Soaking the clothes into water softens the dirty, which is then removed by the addition of movement and friction. Cleaning substances, such as soap and detergent, are added to the water to help the cleaning process, they break up the fatty and dirty molecules chains, dissolving these substances in water (Kallen, 2019). According to Easter, Baker, and McQuerry (2013), heating the water improves the efficiency to remove stains, however, it impacts energy consumption and may lead to clothes damaging.

The dirty and soapy water is removed from the clothes by rising it in clean water. Adding movement and friction improves cleaning efficiency. The excess of water is removed by applying pressure or by centrifugal forces. Finally, the drying process might be through an artificial source of heat, or by natural air drying.

5.2.2. Soap and softener

In order to reduce the surface tension of the water and help the removal of oil, grease, and soil from clothes, soaps and detergents are added to the water, loosening the dirty and suspending it on the water. Soaps are made of fats and oils treated chemically by an alkali substance, usually caustic soda or caustic potash. There are two major processes to produce soap, the first and most common is the saponification, where the fats and oils are heated with the alkali substance, resulting in soap, water, and glycerine. The glycerine is a by-product that is removed from the manufacturing process and can be used for other applications such as food, cosmetics, and drug industries. The second major process to obtain soap is by neutralizing fatty acids with alkali products. In this process, initially, the fats and oils are converted to fatty acids by high-pressure steam (hydrolysis). A distillation process is used to remove unwanted substances from the fatty acids, and finally, soap and water are obtained by the neutralization with the alkali substance (SDA, 1994).

For laundry purposes, it is possible to find soap as liquids, powders, bars, gels, and sticks. The most common for washing machines are the liquids and powders forms. According to (McCoy, 2019), the volume of powder soaps sold around the world in 2018 was approximately double than the liquid form. Powder soaps are more common in Africa, India, China, and Latin America, whereas liquid soaps are the majority in the US.

Liquid soaps have the advantage to be more soluble in water and easier to handle. They are more efficient to remove dirty, however, the production of liquid soap is more complex because all the ingredients have to be compatible with each other and stable during the life cycle of the product. On the other hand, powder soap is easier to produce since each element can be produced independently and then the discrete particles of each substance are mixed. Therefore, powder soap is cheaper and more common in developing countries, even though it's cleaning capacity is lower than the liquid form (McCoy, 2019).

Fabric softeners are added during the rinse part of the washing cycle. These components reduce the wrinkling and make the clothes feel softer by the deposition of lubricating products on the fabric. They are composed of conditioning agents, emulsifiers, and other ingredients. The conditioning agents have an oily nature and are responsible to lubricate and soften the fibers. The emulsifiers are used to create a stable mixture and to avoid the separation of the liquid and conditioning agents in two phases. Finally, ingredients are added to provide fragrance, color, and preserve the product's quality. (How products are made, 2020). Since the softener is not compatible with the chemicals of the soaps, the softener has to be added to the clothes just after the soap is removed in the rinse cycle.

5.2.3. Kinds of fabrics

Several kinds of different fabrics are used as clothing materials. The two major groups are related to the source of the raw materials, which are natural and synthetic cloth. Within the natural group, the most commons fabrics are cotton, cellulosic fibers, wool, silk, and leather. The synthetic group is composed, mainly, of polyester, but also includes nylon, acrylic, elastane, and polypropylene.

Regarding wash cycles, some fabrics are delicate and need a gentle washing procedure. For instance, silk and wool articles have to be washed in a water temperature lower than 40°C with smooth movements. To protect the garment from abrasive damage, it is recommended to not share the load with other kinds of fabrics, such as jeans. Moreover, specific soaps are available for delicate clothes (TIDE, 2020). Before drying, the excess of moisture can be removed by a towel. Natural drying is preferable than heat drying, however, direct contact with sunlight should be avoided (Dove, 2020).

Most fabrics are made of cotton or a blend of cotton with synthetic cloth. These fabrics are more resistant and easier to wash in automatic machines. Depending on the thickness of the garment, more agitation can be used, for instance, when washing jeans clothes. For 100% cotton articles, it is recommended to use ambient temperature water and natural dry to avoid shrinking. The addition of synthetic cloth to cotton makes the fabric more durable, resistant to wrinkles and shrinking, therefore, these fabrics can be washed under a wide variety of conditions, including light, moderate and high amount of agitation, cold or heated water, and natural or heated drying process (Dove, 2020).

5.3. ANALYSIS OF EXISTING PRODUCTS

The first automated washing machine was introduced by Alva J. Fisher in 1907, it was named “Might Thor” and the main difference from previous washing machines is that it was the first to be powered by an electric motor. However, only after World War II, the demand for house appliances has significantly increased, which includes the demand for washing machines (Bristow, 2015). Nowadays, a washing machine is considered a “must-have” appliance for any family, since it is a tremendous time saving and reduces the handwork.

Nowadays there are basically two categories of washing machines, the top-loading and the front-loading. Each one has characteristics that make them suitable for specific groups of consumers. Some variations such as the “twin tube” and the “hybrid” version might also be found in the market, however, they are less common than the top and front-loading types.

5.3.1. Top loading washing machines

The top loading machines are composed of a vertical basket and a water-retaining tube. The basket is covered by a metal sheet that is responsible for structural purposes. The top part is composed of a plastic panel and a hinged lid. The clothes are loaded from the top lid to the basket. An example of a top loading machine is presented in Figure 40.

The mechanism to agitate the water might be either an agitator or an impeller. The agitator is a central blade which has oscillating movements, providing circulating patterns for the water and clothes inside the basket. It is an efficient way to remove dirt from clothes since it has a big contact surface, however, it requires an oscillating mechanism such as a reversible motor or a gearbox and it reduces the available space for dirty clothes.

The impeller has a geometry of a disc with small blades, it spins in the same direction, creating a fast-moving current of water. It has the advantage of mechanical simplicity compared with the agitator, and no oscillatory movement is required. Another advantage is the available space, which is bigger due to the simplicity of the impeller geometry. However, the efficiency to remove dirty is compromised, since it has a lower contact surface, which implies in increased use of cleaning products, such as detergent and soap

Figure 40 - Top loading washing machine.
Font: https://www.winningappliances.com

The main advantage of top loading washing machines is the facility to contain water inside the basket since front loading washing machines require front door sealing. The high-velocity spin is also easier because the vertical position of the basket results in a balanced distribution of the clothes, whereas in front-loading washing machines the clothes accumulate in one side of the basket, creating an unbalanced weight distribution, which requires heavyweights to stabilize the centrifugal process.

5.3.2. Front loading washing machines

In this concept, the basket is mounted in a horizontal position. The back and forth spinning of the basket creates a tumbling movement, producing the necessary agitation and friction to clean the clothes. Front-loading machines do not have an agitator or impeller, instead, small paddles are present in the inner part of the basket, they are responsible for lifting the clothes during the rotational movement and then dropping them.

The mechanical transmission is much simpler, usually, it is composed of a pulley coupling, without the need of a gearbox, clutch, or crank, which are quite common in top-loading machines. However, the main challenges with this technology are related to the horizontal position of the basket. The first challenge is ensuring a proper seal on the front door to prevent water leakage. A special gasket is used for this purpose; however, it may wear out over time. Additionally, the door must remain locked during washing to prevent accidental openings. The second challenge is the cantilever mounting of the basket, which generates bending forces on the bearings. This can lead to bearing wear over time and result in high maintenance costs. Finally, the centrifugal process is also delicate, as the weight distribution is naturally unbalanced, requiring stabilizing weights to ensure the machine remains stable.

Figure 41 - Front loading washing machine.
Font: https://www.us-appliance.com/

The main advantages are related to water consumption, which is significantly reduced because the clothes don't need to be freely suspended in water since the tumbling movement forces the water to moist the clothes. The noise is lower than top loading machines because the door sealing works as noise insulation.

5.3.3. Hybrid washing machines

There are some variations of the traditional top and front-loading washing machines, one example is the twin tub version (Figure 42), which is a simplified version of a top loading machine. It is composed of two tubes, in which one is where the clothes are washed and the second is where the clothes are centrifugated. The washing process is not as automatic as a conventional top-loading machines, however, the mechanical construction is much less complicated, reducing the final price of the product.

Figure 42 - Twin tube washing machine.
Font: https://www.costway.com/

Another example is a model of the front-loading machine where the actual loading is made through a liftable door on the horizontal basket (Figure 43). Even with a top loading procedure, these machines work just like a conventional front-loading machine, with a rotating drum in a horizontal axis. The main advantage is the symmetrical fixation points, which can be done since there is no front door, thus the cantilever effect of conventional front-loading machines can be eliminated.

Figure 43 - Horizontal top loading washing machine.
Font: https://laundry-alternative.com/

5.4. REQUIREMENTS FROM STANDARDS

Three standards provide requirements for commercial washing machines. Most of these requirements are highly specific to the embodiment and detailed design phases (for example, specifications for internal wiring). However, a few requirements may significantly influence the product concept. Therefore, this preliminary analysis of the standards focuses on those that impact the conceptual design.

IEC 60335-2-7

This standard deals with common hazards present in household appliances with rated voltage less than 250 V, specifically, electric washing machines. The main requirements that can be extracted are:

- Protection against moving parts
- Moisture resistance
- Mechanical stability
- Resistance to heat and fire
- Resistance to rusting

IEC 60730-1

This standard presents general requirements for electrical controls in household appliances. The common requirements that are applicable at this early stage of the design process are the same as the IEC 60335-2-7.

IEC 60456

This standard presents methods to determine the efficiency of the washing machines, including washing performance, water extraction performance, rinsing performance, water, and energy consumption. These tests have to made after the product is finished and ready to enter the market, or as a benchmark to compare with the competitors' products and redesign/improve the company's product. At this point, no design requirement can be extracted from this standard.

5.5. PRODUCT IDEA

The main product idea is to develop a household appliance, focusing on middle-income families. The product needs to be efficient in removing the dirt from the clothes, easy to use, and capable to handle the washing processes by itself. Moreover, it needs to preserve clothes' integrity and have a quick cycle time. It has to be able to wash common clothes made of cotton and polyester, delicate silk and wool clothes, and thick and heavy items, such as blankets and jeans. The company has a technology background in the development of automated household appliances, such as blenders, fans, food mixers, etc. The shapes and dimensions have to be similar to the products already present in the market. The manufacturing processes have to consist only of the traditional methods and the product has to be made of common materials widely available in the market. The final product has to be robust, nice looking and the main market strategy is the product price, which has to be considerably lower than the competitors.

5.6. USER REQUIREMENTS

Based on the analysis of existing products, the applicable standards, and the product idea, it is possible to define the user requirements, which are summarized expressions of the qualitative requirements identified in the previous section. Table 5 presents the list of the identified user requirements and their category.

Table 5 - User requirements of the washing machine

Client/User	Category	User necessities	User requirements
User	Functionality	The product needs to be efficient in removing the dirt from the clothes	Remove dirty
		The product needs to preserve clothes integrity	Don't damage the clothes
User	Time	The product needs to have a quick cycle time	Fast operation
User	Usability	The product needs to be easy to use	Easy operation
		It has to be able to wash common clothes made of cotton and polyester, delicate silk and wool clothes, and thick and heavy items, such as blankets and jeans	Properly operation
User	Automation	It needs to be capable to handle the washing processes by itself	Autonomous operation
User	Transportability	The shapes and dimensions have to be similar to the products already present in the market	Easy of transportation
			Suitable dimensions
User	Aesthetics	The product has to be nice looking	Nice look
User	Robustness	The product has to be robust	Be robust
User	Costs	The main market strategy is the product price, which has to be considerably lower than the competitors	Minimal acquisition cost
		Household appliance, focusing on middle-income families	Economical operation
Client	Manufacturing	The manufacturing processes have to consist only of the traditional methods	Simple manufacturing processes
		The product has to made of common materials widely available in the market	Standardized materials
User	Safety	Protection against moving parts	No movable parts exposed

5.7. REQUIREMENTS LIST

The transformation of user requirements to design requirements is a process that assigns defined quantities to the user requirements. It is the transformation of an abstract requirement into engineering characteristics that can be quantified. For instance, the necessity to remove dirt from clothes can be translated to a necessity to produce agitation for, at least, 30 minutes. The abstract requirement of minimal acquisition cost is converted to a manufacturing cost of $1.000,00.

The complete list of the design requirements is presented in the requirements matrix (Figure 44), which also presents the correlation between user requirements and design requirements, and classifies each design requirement according to its importance, which is numerically given by the relative weight (W_R).

As can be seen, the most important design requirement is the manufacturing cost, which is related to the final price of the product, and was the most important requirements from the XYZ company.

The set of classifications, the relative and absolute weights, the value and the verification method of each design requirement results in the requirements list (Figure 44). This information will help the decision process during the selection of technological and working principles of the following steps of the design process. This way, the first phase of the design process is completed.

USE CASE: AUTOMATIC WASHING MACHINE

User Requirements (What)	1 Remove dirt	2 Don't damage the clothes	3 Fast operation	4 Easy operation	5 Properly operation	6 Automous operation	7 Ease of transporttation	8 Suitable dimensions	9 Nice design	10 Be robust	11 Minimal acquisition cost	12 Economical operation	13 Simple manufacturing processes	14 Standardized materials	15 No movable parts exposed	Requirements list						
IF	4	4	3	2	4	3	2	3	2	4	4	3	3	3	3	**Design requirement (how)**	**Demand or Wish**	**Absolute weight (W_A)**	**Relative weight (W_R)**	**Technical/ Economic classfification**	**Values (How much)**	**Verification Method**
1	0	1	2	0	2	0	2	2	1	2	2	2	1	0	0	Adequate load capacity	D	55	3	T	10 kg	Scale
2	1	2	0	0	1	0	0	0	0	0	1	2	0	0	0	Do not use hot substances	D	26	2	T	Ambient temperature	Visual inspection, thermometer
3	2	2	2	0	2	0	0	0	0	1	1	1	0	0	0	Provide agitation	D	41	2	T	30 minutes of agitation	Stopwatch
4	0	0	0	2	0	1	2	2	2	0	0	0	1	0	1	Adequate operation height	W	27	2	T	1~1.5 meters	Measuring tape
5	0	0	0	0	0	0	0	0	0	2	2	0	1	2	0	Resistant to corrosion	D	25	2	T/E	Use galvanized or polyester material	Components datasheet, visual inspection
6	0	0	0	0	0	0	2	2	1	1	2	0	1	1	1	Low weight	D	33	2	T/E	40 kg	Scale
7	0	0	1	1	0	2	0	0	0	0	0	2	0	0	0	Different load operation	W	17	1	T	4 different load cycles	Visual inspection
8	2	1	2	2	1	2	0	0	0	0	1	1	0	0	1	Automatic components	D	42	3	T	Fully automated operation	Visual inspection
9	2	1	2	0	2	0	1	0	0	2	2	2	0	0	0	Adequate power	D	50	3	T	For up to 10 kg of clothes	Measure the motor current consumption
10	1	1	0	0	2	0	2	2	1	2	2	1	1	1	0	Correct sizing	D	53	3	T	For up to 10 kg of clothes	Scale and Measuring tape
11	0	2	0	0	0	0	2	0	2	0	1	0	0	1	0	Rounded surfaces	D	23	1	T/E	No sharp edges	Visual inspection
12	1	0	2	1	1	2	0	1	2	2	2	1	2	2	2	Low manufacturing cost	D	66	4	E	$1,000	Commercial invoice
13	0	0	0	0	0	0	0	1	2	1	2	0	2	2	0	Low manufacturing time	D	31	2	E	36 h	Measure production time
14	2	0	2	1	1	2	0	0	0	0	0	2	0	0	0	Low operation cost	D	32	2	E	$5/cycle	Measure power and water consumption
15	0	0	0	0	0	0	1	0	0	2	0	2	1	2	0	Low maintenance cost	D	25	2	T/E	$50/year	Maintenance planning
16	0	0	0	1	1	2	1	1	2	0	1	0	0	0	2	Protection of moving parts	W	31	2	T	Protect all moving parts	Visual inspection
17	2	0	1	0	2	0	2	1	1	2	1	0	0	0	0	Mechanical stability	D	40	2	T	No translation movement due to vibration	Visual inspection
18	0	0	0	0	1	0	0	0	0	2	2	2	0	2	0	Resistant to heat and fire	D	32	2	T	Insulated supports of heat-generating components	Components datasheet
Status	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**							

Figure 44 - Requirements matrix of the washing machine

5.8. FUNCTIONAL DECOMPOSITION AND MEANS SYNTHESIS

The next step is the development of the concept of the product based on the design requirements. It starts by identifying a global function of the system, which is then decomposed in subfunctions and technical functions. The Functions-Means tree is a functional decomposition tool that aims the research of possible means capable to perform a specific task (function), which increases the possibility to develop an innovative concept. Therefore, it was used to decompose the global function of the washing machine into operational means (Figure 45).

The global function is clearly defined as “wash clothes”, which is the main function of a washing machine. There are two means that can be used to wash clothes, water wash, and dry wash. To decide which one is more adequate according to the design specifications, an evaluation chart is used (Table 6), where a qualitative comparison is performed between the two means and the design requirements. It is important to note that a few design requirements (such as “load capacity”) were not considered in this analysis because they are not relevant for both means that are being analyzed. The focus is on the design requirements that have a different impact in each mean, for instance, the resistance to corrosion has a lower weight for the dry wash (W_{S2}=2) because it uses chemical solvents, therefore this mean is more susceptible to corrosion than the water wash (W_{S1}=3). The selection of a mean is based on the highest hyperbolic rating, which is, in this case, the “water wash” mean.

Table 6 - Evaluation chart of level 1

Evaluation Chart								
	Design Criteria	j=	1		2		o	
		W_r	Water Wash		Dry wash		Optimal	
			W_{s1}	$W_{s1}.W_r$	W_{s2}	$W_{s2}.W_r$	W_o	$W_o.W_r$
i=	Technical analysis							
1	Provide agitation	2	4	8	3	6	4	8
2	Resistant to corrosion	2	3	6	2	4	4	8
3	Low weight	2	4	8	2	4	4	8
	Total technical value		T_{t1}	**22**	T_{t2}	**14**	T_{to}	**24**
	Relative technical value		R_{t1}	**0.92**	R_{t2}	**0.58**	R_{to}	**1.00**
i=	Economic analysis							
4	Resistant to corrosion	2	4	8	2	4	4	8
5	Manufacturing cost	4	4	16	2	8	4	16
6	Operation cost	2	4	8	2	4	4	8
7	Maintenance cost	2	3	6	2	4	4	8
	Total economic value		T_{e1}	**38**	T_{e2}	**20**	T_{eo}	**40**
	Relative economic value		R_{e1}	**0.95**	R_{e2}	**0.50**	R_{eo}	**1.00**
	Hyperbolic rating		H_{r1}	**0.93**	H_{r2}	**0.54**	H_{ro}	**1.00**

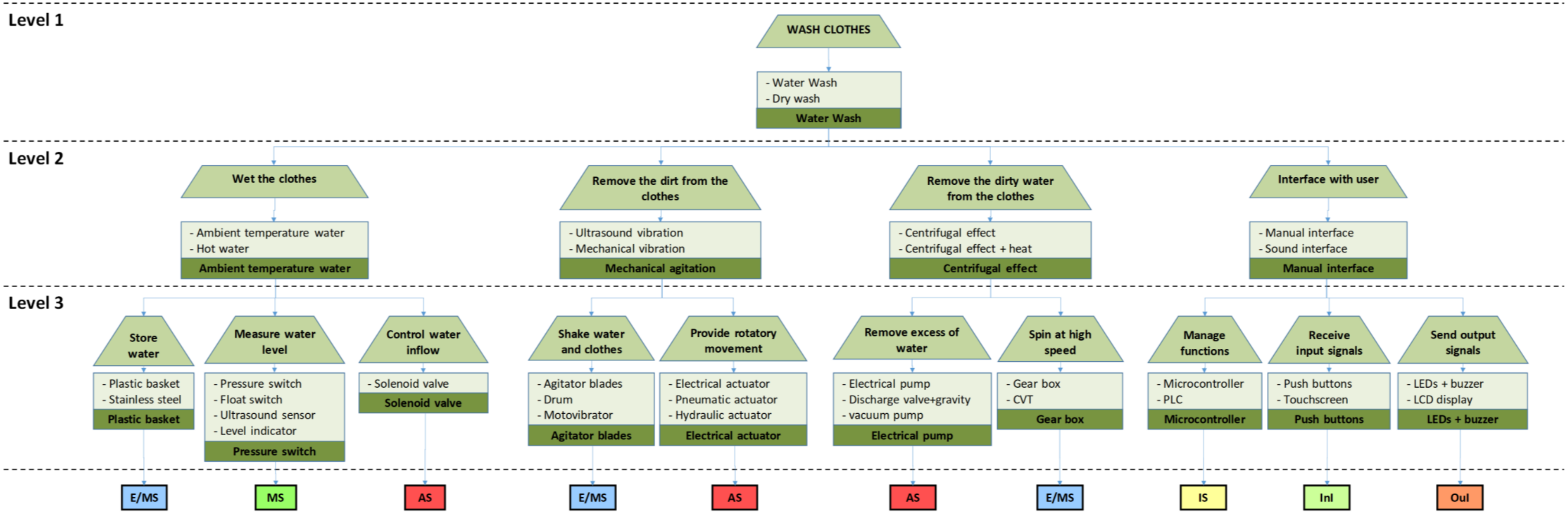


Figure 45 - Functions-Means tree of the washing machine

The selected mean (water wash) was decomposed in 4 subfunctions. For each subfunction, two possible means were defined. A morphological matrix was used to generate possible solutions through the combination of the subfunctions means. As can be seen in Figure 46, three possible solutions were proposed. To decide which one best meets the design requirements, an evaluation chart (Table 7) was used along with the design requirements relevant to these possible solutions. According to the hyperbolic rating, solution 1 was selected.

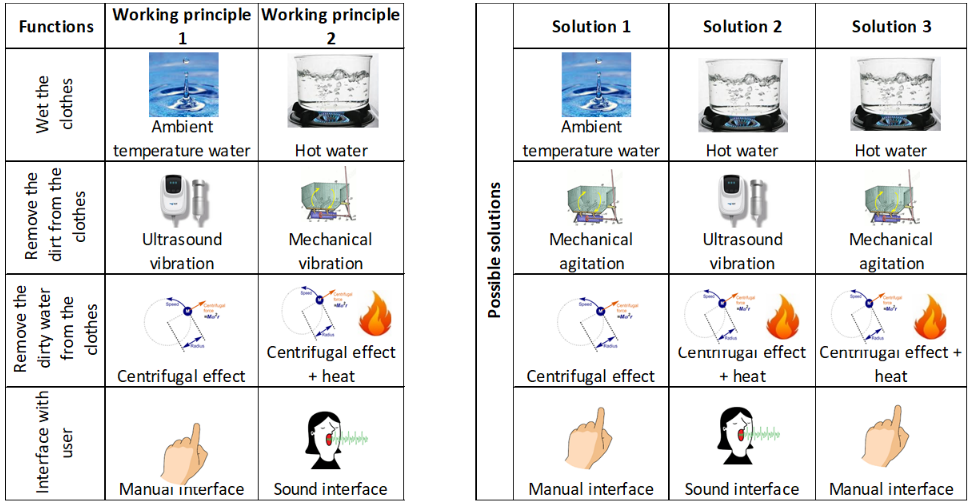


Figure 46 - Morphological matrix of level 2

Table 7 - Evaluation chart of level 2

Evaluation Chart										
	Design Criteria	j=	1		2		3		o	
		W_r	Solution 1		Solution 2		Solution 3		Optimal	
			W_{s1}	$W_{s1}.W_r$	W_{s2}	$W_{s2}.W_r$	W_{s3}	$W_{s3}.W_r$	W_o	$W_o.W_r$
i=	Technical analysis									
1	Do not use hot substances	2	4	8	0	0	0	0	4	8
2	Provide agitation	2	4	8	3	6	4	8	4	8
3	Resistant to corrosion	2	3	6	4	8	3	6	4	8
4	Low weight	2	3	6	4	8	3	6	4	8
5	Rounded surfaces	1	3	3	4	4	3	3	4	4
6	Protection of moving parts	2	3	6	4	8	3	6	4	8
7	Mechanical stability	2	3	6	4	8	3	6	4	8
8	Resistant to heat and fire	2	4	8	2	4	2	4	4	8
	Total technical value		T_{t1}	51	T_{t2}	46	T_{t3}	39	T_{to}	60
	Relative technical value		R_{t1}	0.85	R_{t2}	0.77	R_{t3}	0.65	R_{to}	1.00
i=	Economic analysis									
9	Resistant to corrosion	2	4	8	3	6	4	8	4	8
10	Low weight	2	4	8	3	6	4	8	4	8
11	Manufacturing cost	4	4	16	3	12	3	12	4	16
12	Manufacturing time	2	3	6	4	8	2	4	4	8
13	Operation cost	2	4	8	4	8	2	4	4	8
14	Maintenance cost	2	4	8	3	6	3	6	4	8
	Total economic value		T_{e1}	54	T_{e2}	46	T_{e3}	42	T_{eo}	56
	Relative economic value		R_{e1}	0.96	R_{e2}	0.82	R_{e3}	0.75	R_{eo}	1.00
	Hyperbolic rating		H_{r1}	0.91	H_{r2}	0.79	H_{r3}	0.70	H_{ro}	1.00

The selected means of level 2 were decomposed in subfunctions, resulting in 10 subfunctions for the third level of the F-M tree. Analyzing the morphological matrix (Figure 47) of the third level, three possible solutions were proposed. It's important to notice that a few means (motovibrator, for instance) were not used in any solution because there is no viable combination of this mean with the other means. However, during the definition of possible means of a subfunction, it's important to consider all means, because it increases the probability to develop an innovative concept.

Table 8 presents the evaluation chart that was used to perform a qualitative analysis between the possible solutions and the relevant design requirements. Solution 2 was chosen because of its highest hyperbolic rating, meaning that it is the solution that will best attend the design requirements.

The selected means of level 3 are components commercially available. The further decomposition of such components is not relevant to the conceptual design of this product because they already perform elementary functions. Therefore, these means are denominated operational means and they are classified according to the architecture of a mechatronic system (energy/material system, actuation system, measurement system, information system, and input/output interface).

Since all the means of the F-M tree are classified as operational means, there is no need to continue the decomposition of the means in subfunctions. The functional decomposition and means synthesis is concluded.

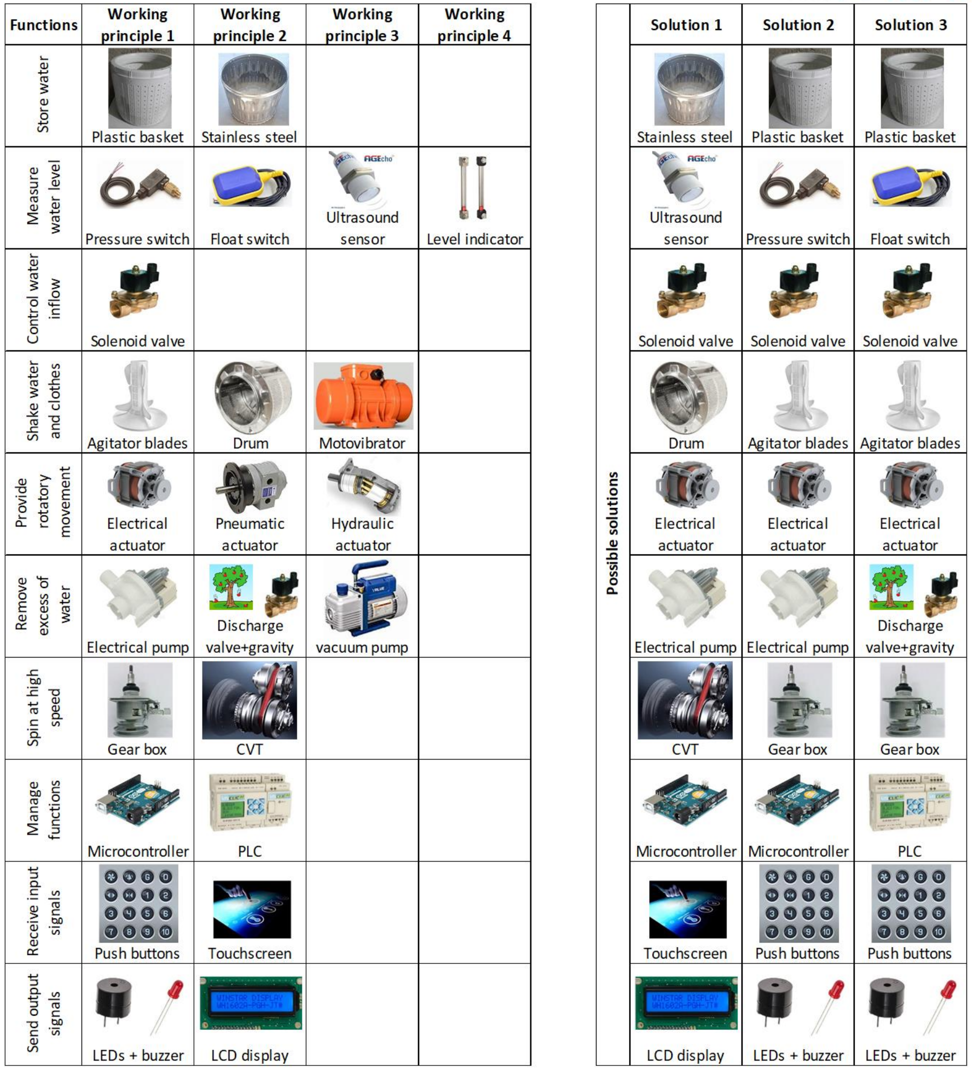


Figure 47 - Morphological matrix of level 3

Table 8 - Evaluation chart of level 3

Evaluation Chart										
		j=	1		2		3		o	
	Design Criteria	W_r	Solution 1		Solution 2		Solution 3		Optimal	
			W_{s1}	$W_{s1}.W_r$	W_{s2}	$W_{s2}.W_r$	W_{s3}	$W_{s3}.W_r$	W_o	$W_o.W_r$
i=	**Technical analysis**									
1	Load capacity	3	4	12	3	9	3	9	4	12
2	Provide agitation	2	3	6	4	8	4	8	4	8
3	Resistant to corrosion	2	4	8	3	6	3	6	4	8
4	Low weight	2	2	4	4	8	4	8	4	8
5	Different load operation	1	4	4	4	4	2	2	4	4
6	Automatic components	3	4	12	4	12	1	3	4	12
7	Mechanical stability	2	2	4	4	8	4	8	4	8
	Total technical value		T_{t1}	**50**	T_{t2}	**55**	T_{t3}	**44**	T_{to}	**60**
	Relative technical value		R_{t1}	**0.83**	R_{t2}	**0.92**	R_{t3}	**0.73**	R_{to}	**1.00**
i=	**Economic analysis**									
8	Resistant to corrosion	2	2	4	4	8	4	8	4	8
9	Manufacturing cost	4	2	8	4	16	4	16	4	16
10	Manufacturing time	2	2	4	3	6	4	8	4	8
11	Operation cost	2	3	6	3	6	3	6	4	8
12	Maintenance cost	2	2	4	3	6	4	8	4	8
	Total economic value		T_{e1}	**26**	T_{e2}	**42**	T_{e3}	**46**	T_{eo}	**48**
	Relative economic value		R_{e1}	**0.54**	R_{e2}	**0.88**	R_{e3}	**0.96**	R_{eo}	**1.00**
	Hyperbolic rating		H_{r1}	**0.67**	H_{r2}	**0.90**	H_{r3}	**0.84**	H_{ro}	**1.00**

5.9. MECHATRONIC SYSTEM ARCHITECTURE

Now that all the main components of the washing machine were selected, it is possible to organize them in a structured form and develop representations for each specific domain: mechanical, electrical, and software.

The first representation is the mechatronic system architecture, which groups all the operational units according to their classification and connects them through the resource that is flowing from/to each operating mean. The mechatronic system architecture helps to visualize and comprehend the working of the whole system since all main components are linked to each other.

Six resources are entering in the system, a signal (user pressing button), electrical energy for the actuation system, auxiliary energy to supply low voltage energy for the electrical components, dirty clothes and soap that go into the basket, and water, that pass through the solenoid valve and enters the basket. The outputs of the system are the visual and sonorous signals, water mixed with soap and dirt, and clean clothes.

The input/output interfaces are working as information converters, they receive one type of signal and convert to another type, for instance, the push buttons receive a mechanical signal from the user and convert it to an electrical signal for the microcontroller. Similarly, the measurement system (pressure switch) receives a column of water (water and pressure) and converts it to an electrical signal. The microcontroller receives input information from the external environment and from the pressure switch, processes this information and sends the outputs for the actuation system and the external environment.

The actuation system receives information from the microcontroller and electrical energy from the external environment. They are either processing material (electrical pump and solenoid valve) or supplying energy (electrical actuator). The energy/material systems receive mechanical energy and use it to perform work on the materials (water, clothes, and soap).

As can be seen in Figure 48, the mechatronic system architecture presents a clear understanding of how the information, energy, and materials are flowing inside the washing machine. This understanding is essential for the construction of a mechatronic system because it involves three different technology domains (mechanical, electrical, and software) that have to be properly integrated to operate correctly.

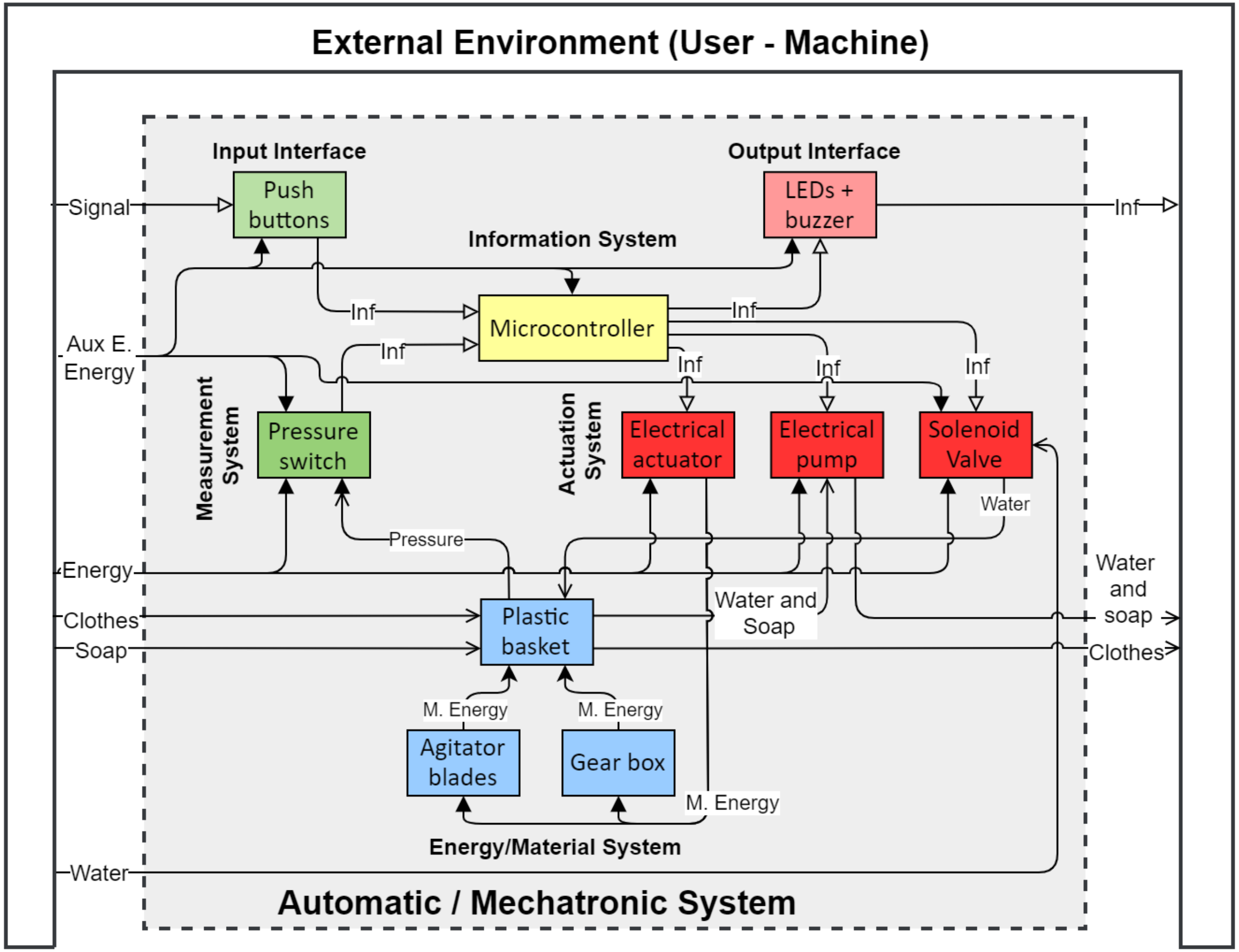


Figure 48 - Mechatronic system architecture of the washing machine

5.10. GENERAL PHYSICAL VIEW

The general physical view is the physical and structural organization of the components. It involves the creation of a structure that will hold all the components in place, and the external shape of the product.

As can be seen in Figure 49, a top-loading model is being proposed, which is due to the agitation mechanism selected based on the design requirements. The gear-box, the electrical actuator, and the electrical pump are located under the plastic basket, which is held by four sustaining cables. The agitation blades are placed inside the plastic basket, connected to the gear-box. The control panel will be installed on the top of the machine, with the push buttons, the LEDs, and the buzzer. A water-prove case will hold the electronic components. The water inflow will pass through the soap dispenser, located below the control panel, and fall into the plastic basket. Transporting handles will be installed in the lateral of the machine. The water output hose is located at a higher level than the plastic basket to avoid leaking. A hose connects the bottom of the plastic basket with the pressure switch.

As requested by the XYZ company, the external shape of the washing machine is similar to what is already present in the market. Moreover, a few design requirements are preliminary being met, such as: do not use hot substances, provide agitation, operation height, rounded surfaces, protection of moving parts, and mechanical stability.

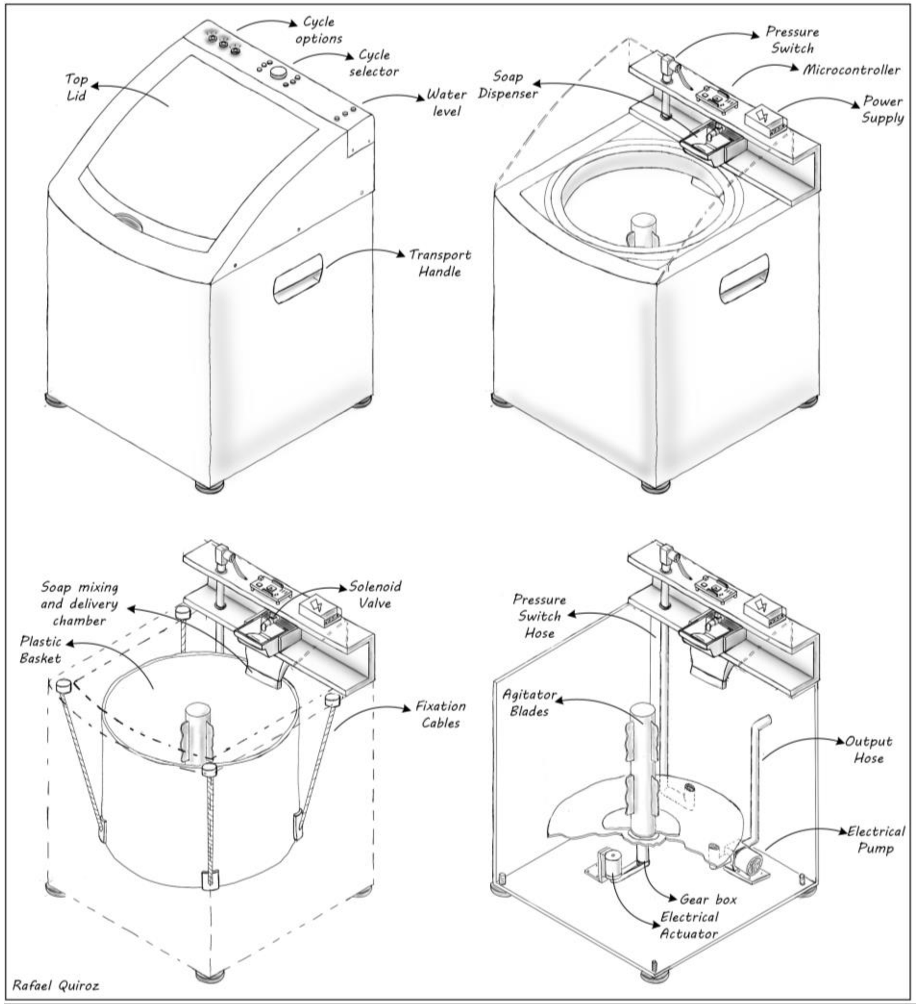


Figure 49 - General physical view of the washing machine

The preliminary power diagram (Figure 50) describes the hydraulic circuit, which is composed, in this case, by an electrical pump, a solenoid valve, an electric motor and a pressure switch. As can be seen, this diagram is focused on a schematic representation of the actuators and sensors, as well as their interconnection inside the washing machine.

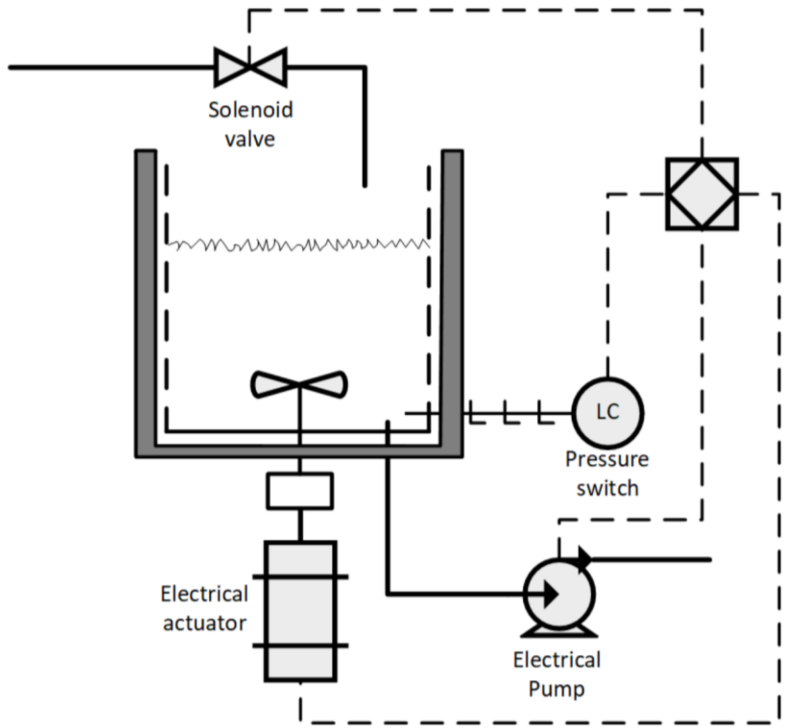


Figure 50 – Water hydraulic diagram of the washing machine

5.11. THE ARCHITECTURE OF THE ELECTRIC/ELECTRONIC SYSTEM

The architecture of the electric/electronic system is a block diagram used to present more details about the structure of the electrical components present on the system. The objective is to identify the power source and how it is associated with the input/output interface, the actuation, measurement, and information system. Moreover, there is a refinement of the mechatronic system architecture, where the main categories are expanded to indicate which components are necessary to perform such a task and the specific functions of these components.

As can be seen in Figure 51, the input signals are given by a cycle selector, a water level selector, and three push buttons (turn on/off the washing machine, select a double rinse cycle, and to advance to the next washing step). The output signals consist of a buzzer and a set of nine LEDs that indicate the current status of the washing machine: Turned on/off, water level (low, medium, high), current step (filling, washing, rinsing, centrifuging), and the double rinse option (yes/no).

The actuation module is composed of the control relays and three actuators (solenoid valve, electric motor, and electric pump). The measurement module has just the pressure switch. The power supply module has an on/off switch, a switching power supply that feeds the DC components of the system, and DC and AC power outputs. The connecting lines indicate the flow of energy and information throughout the system.

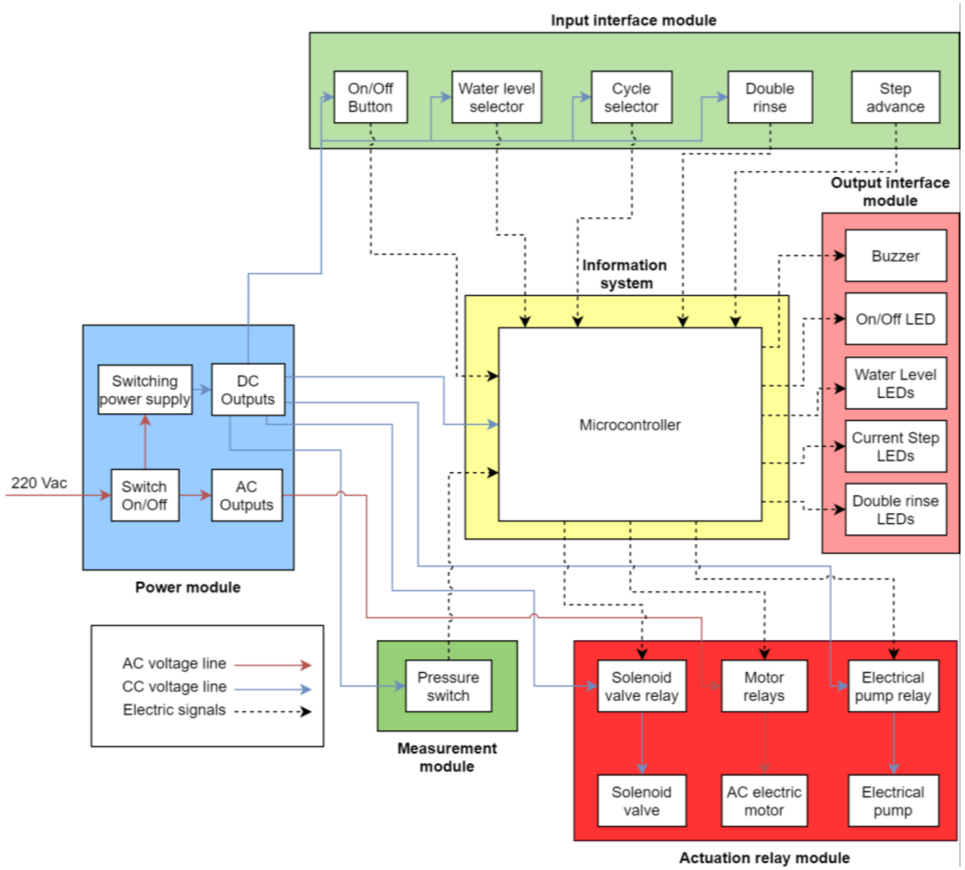


Figure 51 - Architecture of electric/electronic system of the washing machine

5.12. BEHAVIORAL DIAGRAM

The desired washing cycle will determine system behavior, therefore, it can be represented by a flowchart with the stages of the washing cycle and the actions required for each stage. The flowchart is helpful to develop the controlling algorithms because it gives a clear understanding of the system behavior.

As can be seen in Figure 52, the washing process starts with the user inputs, where the water level, washing cycle, and the double rinse are selected. The next step is the water filling, which is controlled by the solenoid valve. During the washing stage, the agitator blades oscillate back and forth, which is caused by the gear box and the counterclockwise rotation of the electric motor. After a specific washing time, the dirty water is removed by the electrical pump. To rinse the clothes, it is necessary to fill the basket with clean water and agitate it during a period of time. If the double rinse option was selected, it is necessary to drain the water, refill the basket with clean water and agitate again. To centrifugate the clothes, the electrical pump is used to drain the excess of water, and finally, the basket spins at high velocity due to the clockwise rotation of the electric motor.

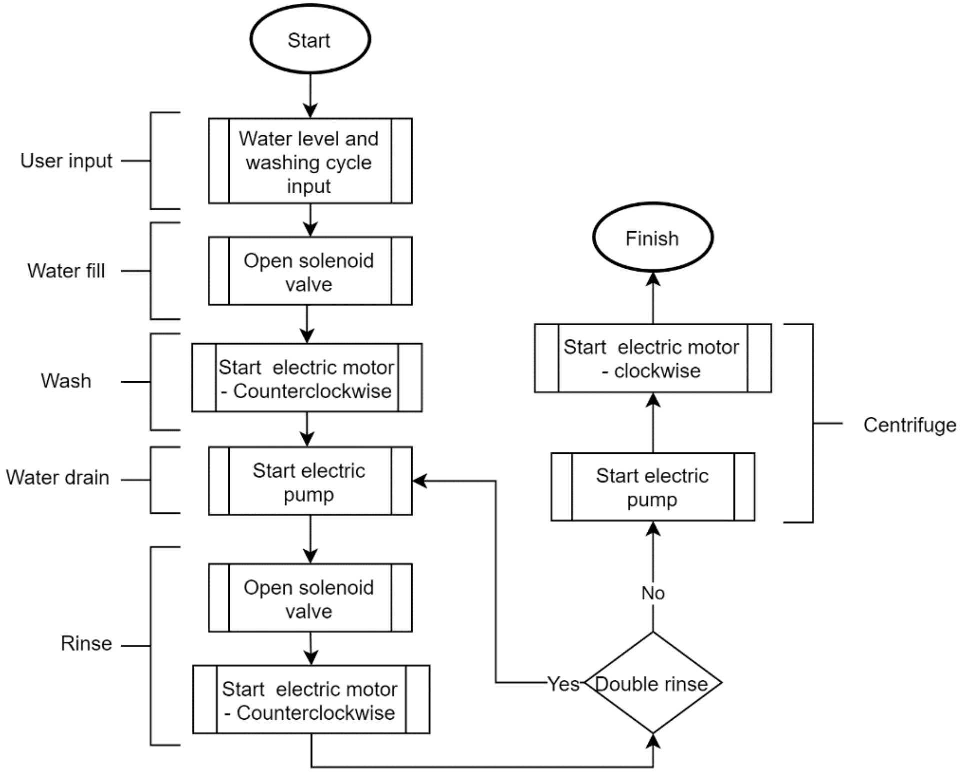


Figure 52 - Behavioral diagram of the washing machine

5.13. SOFTWARE REQUIREMENTS SPECIFICATIONS

The software requirement specifications are a complement of the behavioral diagram. It describes with more details the working of the system and defines values for the variables, such as times, velocity, water level, and washing cycle. The variables are organized in system classes, which helps the implementation of the software.

System characteristics

At the start of the water machine's cycle, the software waits for 10 seconds for a user input update, after this time, the variables are filled according to Table 9, Table 10 and Table 11, which relates the user input to the system operation variables. During operation, the software will control the electric motor, solenoid valve, and the electric pump through the respective component's relay. The objective of the software is to activate and deactivate the component's relay according to the selected program and send feedback signals to the user. The step advance signal will jump to the next washing step.

Table 9 - Washing cycle parameters

Washing cycle	Wash time	Wash velocity	Centrifuge time
Quick	10	600	5
Soft	15	300	10
Normal	20	600	10
Heavy	30	800	15

Table 10 - Double rinse parameters

Double rinse	Number of rinse cycles
No	1
Yes	2

Table 11 - Water level parameters

Water level	Water level value	Emptying time
Low	4	2
Medium	8	4
High	12	6

Software functionalities

Five functionalities must be implemented in the software, which will be responsible for operating the actuation system according to the following description:

- **Water filling:** Activate the solenoid valve's relay while the actual water level is lower than the water level value;
- **Washing:** Activate the electric motor relay while the elapsed wash time is lower than the wash time. The rotation direction has to be counterclockwise and rotation speed has to follow Table 9 values. The active time is intermittent with 1 minute off for every 2 minutes on.
- **Emptying:** Activate the electric pump's relay while the elapsed emptying time is lower than the emptying time.
- **Centrifuge:** Activate the electric motor's relay while the elapsed centrifuge time is lower than the centrifuge time. The rotation direction has to be clockwise and the rotation speed is 800 RPM.
- **Coordinator:** Manage the sequence of functions during the washing cycle according to the behavioral diagram.

System classes

There are 5 classes of variables that will be used to share information through the algorithm. The classes are presented at the entity-relationship diagram of Figure 53.

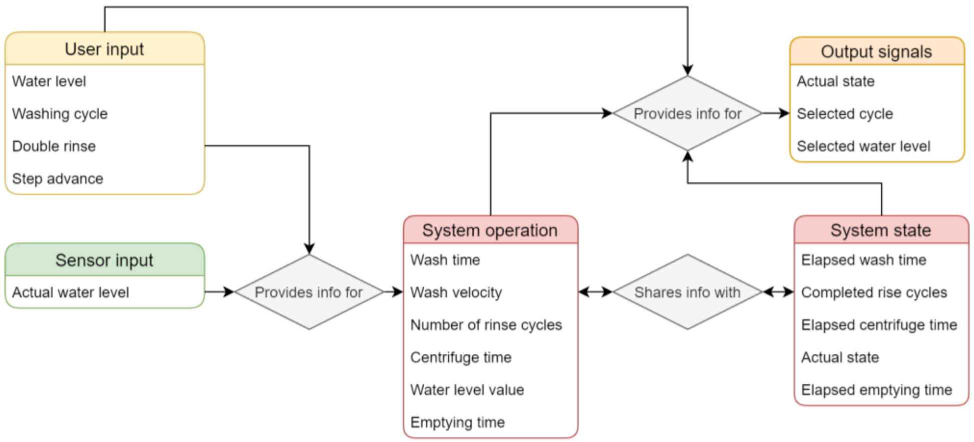


Figure 53 - Entity-Relationship of the washing machine

With the diagrams and sketches developed during the second phase of the design process, it was possible to create a clear concept of the product. The three domains of mechatronic systems are well structured and represented by the general physical view, the electric/electronic diagram, the behavior diagram, and the software requirements specification. The architecture of mechatronic systems integrates these technologies, giving a comprehensive understanding of the correlation between the main parts of the system. Moreover, several technological and/or working principles were proposed and evaluated by the Functions-Means tree, the morphological matrices, and the evaluation charts, resulting in a combination of operational means that best attends the design requirements.

REFERENCES

Andreasen, M. M., Hansen, C. T., & Cash, P. (2015). *Conceptual Design-Interpretations, Mindset and Models, Conceptual Design*. Switzerland: Springer.

Back, N., Ogliari, A., Dias, A., & Silva, J. d. (2008). *Projeto integrado de produtos: planejamento, concepção e modelagem*. Barueri, SP.

Belan, H. C. (2007). *Formalização da rede de Petri canal/agência para projeto de equipamentos industriais.* (Master thesis), Federal University of Santa Catarina, Florianópolis, Brazil.

Belan, H. C., Szpak, R., & De Negri, V. J. (2010). Channel/Instance Petri net for structural and functional modeling of industrial equipment. *ABCM Symposium Series in Mechatronics, 4*, 403-407.

Bristow, M. (2015). Who invented the washing machine? A brief history of laundry technology. Retrieved from https://www.appliancesonline.com.au/academy/appliance-questions/who-invented-the-washing-machine-a-brief-history-of-laundry-technology/

Buur, J. (1990). *A theoretical approach to mechatronics design*. Denmark: Institute for Engineering Design, Technical University of Denmark.

Cassandras, C., & Lafortune, S. (2008). *Introduction to Discrete Event Systems* (2 ed.): Springer US.

Dathan, B., & Ramnath, S. (2015). *Object-oriented analysis, Design and implementation* (Second Edition ed.). New York: Springer.

De Negri, V. J. (1996). *Estruturação da modelagem de sistemas automáticos e sua aplicação a um banco de testes para sistemas hidráulicos.* (Doctorate Thesis), Federal University of Santa Catarina, Florianópolis, Brazil.

De Negri, V. J. (1998). *Conception of automatic test benches for hydraulic components.* Paper presented at the the power transmission and motion control (PTMC'98), , Bath, UK.

De Negri, V. J., & Santos, E. A. P. (2007). Projeto de sistemas de automação da manufatura *Enciclopédia de Automática* (Vol. 1, pp. 382-417). São Paulo, Brazil: Edgar Blucher.

Dove, T. (2020). Fabric, Classification and Care (1 ed.).

Easter, E. P., Baker, E., & McQuerry, M. (2013). *Assessing the impact of wash water temperature, detergent type and laundering platform on basic clothing attributes.* Paper presented at the AATCC International Conference, Greenville, South Carolina.

Frederick, D. K., & Carlson, A. B. (1971). *Linear systems in communication and control*. New York, USA: John Wiley & Sons.

Gajski, D. D., & Kuhn, R. H. (1983). New VLSI tools. *Computer, 16*(12), 11-14.

Gerstlauer, A., Haubelt, C., Pimentel, A. D., Stefanov, T. P., Gajski, D. D., & Teich, J. (2009). Electronic system-level synthesis methodologies. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28*(10), 1517-1530.

Hanisch, H.-M. (1992). *Petri-Netze in der Verfahrenstechnik: Modellierung* *und Steuerung verfahrenstechnischer Systeme*. München: Oldenbourg.

Hansen, C. T., & Andreasen, M. M. (2002). Two approaches to synthesis based on the domain theory *Chakrabarti, A, Engineering design synthesis: Understanding, Approaches and Tools* (pp. 93-108). London: Springer-Verlag.

Henson, B., Juster, N., & de Pennington, A. (1994). *Towards an integrated representation of function, behavior and form.* Paper presented at the 1994 Lancaster International Workshop on Computer Aided Conceptual Design.

Heuser, C. A. (1990). *Modelagem conceitual de sistemas: redes de Petri*. Campinas, Brazil: Escola Brasileiro-Argentina de Informática. R. Vieira.

Hoover, S. P., Rinderle, J. R., & Finger, S. (1991). *Models and abstractions in design.* Paper presented at the International Conference on Engineering Design, Zurich.

How products are made. (2020). Fabric softener. Retrieved from http://www.madehow.com/index.html

Hubka, V., & Eder, W. E. (1988). *Theory of Technical Systems A Total Concept Theory for Engineering Design*. Germany: Springer-Verlag.

IEC. (2013). GRAFCET *Specification language for sequential function charts* (Vol. IEC 60848): International Electrotechnical Commission.

Isermann, R. (2007). *Mechatronic systems: fundamentals*. London, UK: Springer Science & Business Media.

ISO. (2012a). Fluid power systems and components *Graphical symbols and circuit diagrams -- Part 1: Graphic symbols for conventional use and data-processing applications* (Vol. ISO 1219-1): International Organization for Standardization.

ISO. (2012b). Fluid power systems and components *Graphical symbols and circuit diagrams -- Part 2: Circuit diagrams* (Vol. ISO 1219-2): International Organization for Standardization.

ISO. (2016). Fluid power systems and components *Graphical symbols and circuit diagrams -- Part 3: Symbol modules and connected symbols in circuit diagrams* (Vol. ISO 1219-3): International Organization for Standardization.

Jacobson, I., Christerson, M., Jonsson, P., & Overgaard, G. (1994). *Object-oriented Software Engineering: A Use Case Driven Approach*. USA: Addison-Wesley.

Jalote, P. (2005). *An integrated approach to software engineering* (3 ed.). New York: Springer Science & Business Media.

Kallen, L. E. (2019). How does soap actually work? Retrieved from https://www.defeatdd.org/

Karnopp, D., Margolis, D., & Rosenberg, R. (1990). *System Dynamics: A Unified Approach* (2 ed.). New York.

Kumara, S., Ham, I., Al-Hamando, M., & Goodnow, K. (1989). Causal reasoning and data abstraction in component design. *CIRP annals, 38*(1), 145-148.

McCoy, M. (2019). Almost extinct in the US, powdered laundry detergents thrive elsewhere in the world. *Chemical and Engineering News*.

Nygaard, K. (1986). Basic concepts in object oriented programming. *ACM*

SIGPLAN Notices, 21(10), 128-132.

Object Management Group. (2021). Unified Modeling Language (UML) Resource Page. Retrieved from https://www.omg.org/

Paes, F. H. S., & De Negri, V. J. (2003). *Sistematização do Projeto de Automação de Sistemas Mecatrônicos com Aplicação às Pequenas Centrais Hidrelétricas*. Paper presented at the Seminário Nacional de Controle e Automação Industrial, Elétrica e de Telecomunicações, Salvador, Brasil.

Pahl, G., & Beitz, W. (2013). *Engineering design: a systematic approach* (3 ed.). London: Springer Science & Business Media.

Reisig, W. (1985). *Petri nets: an introduction*. Berlin: Springer-Verlag.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. E. (1991). *Object-oriented modeling and design*. New Jersey: Prentice-hall Englewood Cliffs.

Santos, E. A. P., Cury, J. E. R., & De Negri, V. J. (2001). *A computational model for supporting conceptual design of automatic systems*. Paper presented at the Design Research - Theories, Methodologies, and Product Modelling - International Conference on Engineering Design, (ICED 01), Glasgow, UK.

SDA. (1994). Soap and Detergents (2 ed.). Washington: The soap and detergent association.

Simon, H. A. (1981). *The sciences of the artificial*. Cambridge: MIT Press.

Song, I.-Y., Evans, M., & Park, E. K. (1995). A comparative analysis of entity-relationship diagrams. *Journal of Computer and Software Engineering, 3*(4), 427-459.

Svitavy Museum. (2020). History of washing. Retrieved from http://www.muzeum.svitavy.cz/

Teich, J. (2001). Synthesis and Optimization of Digital Hardware/Software Systems. In W. S. Renate Merker (Ed.), *System Design Automation: Fundamentals, Principles, Methods, Examples* (pp. 3-26): Springer.

Teich, J. (2012). Hardware/software codesign: The past, the present, and predicting the future. *Proceedings of the IEEE, 100*.

TIDE. (2020). Washing silk. Retrieved from https://tide.com/

Ullman, D. G. (2016). *The mechanical design process* (6 ed.). New York: McGraw-Hill.

Ulrich, K. T., & Eppinger, S. D. (2016). *Product design and development*. New York, USA: McGraw-Hill Education.

VDI. (1987). Systematic Approach to the Design of Technical System and Products (Vol. VDI 2221, pp. 32). Düsseldorf: The Association of German Engineers,.

VDI. (1997). Design engineering methodics *Engineering design at optimum cost - Simplified calculation of costs* (Vol. VDI 2225 BLATT 1, pp. 24). Düsseldorf: Verein Deutscher Ingenieure.

VDI. (1998). Design engineering methodics *Tables for engineering design at optimum cost* (Vol. VDI 2225 BLATT 2, pp. 37). Düsseldorf: Verein Deutscher Ingenieure.

VDI. (2004). Entwicklungsmethodik für mechatronische Systeme *Design methodology for mechatronic systems* (Vol. VDI 2206, pp. 118). Düsseldorf: The Association of German Engineers.

Wang, G. G. (2002). Definition and review of virtual prototyping. *Journal of Computing and Information Science in Engineering, 2*(3), 5. doi:10.1115/1.1526508

Wikipedia. (2021). Mechatronics. Retrieved from https://en.wikipedia.org/wiki/Mechatronics

Yourdon, E. (1989). *Modern structured analysis*. New York: Yourdon Press.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

EMC-UFSC LASHIP HYDRAULICS & PNEUMATICS

PUCP

https://laship.ufsc.br/
Federal University of Santa Catarina
Technological Center, Department of Mechanical Engineering
LASHIP – Laboratory of Hydraulic and Pneumatic Systems
Campus Universitário, Trindade, 88040-900, Florianópolis, SC, Brazil

ISBN: 978-85-8328-351-5